Winner-Take-All Multiple Category Boosting for Multi-View Face Detection

Cha Zhang and Zhengyou Zhang
Communication and Collaboration Systems Group, Microsoft Research
One Microsoft Way, Redmond, WA 98052
{chazhang,zhang}@microsoft.com

November 2009

Technical Report
MSR-TR-2009-190

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

Abstract

"Divide and conquer" has been a common practice to address complex learning tasks such as multi-view object detection. The positive examples are divided into multiple subcategories for training subcategory classifiers individually. However, the subcategory labeling process, either through manual labeling or through clustering, is suboptimal for the overall classification task. In this paper, we propose multiple category boosting (McBoost), which overcomes the above issue through adaptive labeling. In particular, a winner-take-all McBoost (WTA-McBoost) scheme is presented in detail. Each positive example has a unique subcategory label at any stage of the training process, and the label may switch to a different subcategory if a higher score is achieved by that subcategory classifier. By allowing examples to self-organize themselves in such a winner-take-all manner, WTA-McBoost outperforms traditional schemes significantly, as supported by our experiments on learning a multi-view face detector.

1. Introduction

The state of the art in object detection has made significant progress in recent years. Take face detection as an example, the seminal work by Viola and Jones [21], which relies on rapid Haar features, AdaBoost learning and a cascade structure, has shown satisfactory performance for frontal face detection tasks. On the other hand, when objects are observed from multiple viewpoints, the detection task becomes substantially harder. If all object examples are labeled indifferently as positive examples, the detector learned through a straightforward learning algorithm such as AdaBoost will not perform accurately.

The common practice in multi-view object detection has been "divide and conquer". Namely, the general class of objects is first divided into subcategories. Different classifiers can then be trained for different subcategories. For instance, faces can be categorized as frontal, left/right half profile, left/right profile, 0 degree in-plane rotation, ± 30 degree in-plane rotation, etc. In the face detection work in [10, 8, 23], a pose estimator is first built to classify each example into one of the above subcategories. Each subcategory then trains its own classifier for detection with *manually* labeled data. The manual labeling process is very labor-intensive, and sometimes difficult to do for tasks such as pedestrian detection or car detection. In [18, 24], researchers proposed to obtain these labels via automatic clustering. Take the clustered boosted tree classifier in [24] as an example. They applied a conventional k-means clustering algorithm to split the sample set into two parts when the learning rate slows down. They showed that by using the previously selected features for clustering, the learning algorithm converges faster and achieves better results.

One weakness that exhibited in early works [10, 8, 23] is the misclassification caused by the pose estimator. If a profile face is misclassified as frontal, it may never be detected in later stages. In [5], Huang et al. proposed vector boosting, which allows an example to be passed into multiple subcategory classifiers during testing, and the final results are fused through linear transform. Such a soft branching scheme can greatly reduce the risk of misclassification during testing.

Misclassification also happens in training. It could be caused by mislabeling. For instance, the boundary between frontal and half profile faces can be very subtle, and differs from person to person. For systems that rely on automatic clustering to derive the subcategory labels, misclassification can be very common. The misclassified examples appear as outliers in its designated subcategory, which may hinder the learning process and degrade the classifier performance. More importantly, although the manual labels or clustered results are meaningful for the system designer, there is no guarantee that they are optimal for learning the overall detector. Traditional training processes [5, 24] lack the flexibility to re-categorize examples during training, thus forming updated clusters which can help achieving the optimal performance.

In this paper, we propose *multiple category learning* (MCL), which overcomes the above issue through adaptive labeling. During the learning process, we allow the example subcategory labels to be modified in order to make better object/non-object decision. More specifically, we propose a novel boosting algorithm called winner-take-all multiple category boosting (WTA-McBoost). In our approach, multiple subcategory boosting classifiers are learned simultaneously with the assumption that the final classification of an example will only be determined by the highest score of all the subcategory classifiers, i.e., the winner will take all. The subcategory labels of the examples are dynamically assigned in this process, reducing the risk of having outliers in each subcategory. The WTA-McBoost algorithm uses confidence-rated prediction [16] with asymmetric cost and is thus very efficient to train and test. We demonstrate the effectiveness of WTA-McBoost by building a multi-view face detector, and show its superior performance to traditional approaches.

The rest of the paper is organized as follows. WTA-McBoost is described in Section 2, followed by discussions about McBoost and existing approaches in Section 3. Experimental results are presented Section 4. Conclusions and future work are given in Section 5.

2. Winner-Take-All McBoost

Without loss of generality, consider a two-class classification problem as follows. A set of labeled examples

$\mathcal{S} = \{(x_i, z_i), i = 1, \cdots, N\}$ are given for training, where $z_i = 1$ for positive examples and $z_i = 0$ for negative examples. In order to perform "divide and conquer", let us assume that the positive examples can be classified into $k = 1, \cdots, K$ subcategories, either by manual labeling or automatic clustering. Since the manual labels or the clustering results are not directly optimized for the overall two-class classification task, it would be suboptimal if we train K classifiers separately.

In our approach, we will train K boosting classifiers *jointly*. Recall in boosting each example is classified by a linear combination of weak classifiers. Let $y_{ik}^T = H_k^T(x_i) = \sum_{t=1}^T \lambda_k^t h_k^t(x_i)$ be the weighted sum of weak classifiers for subcategory k, often referred as the *score* of classifier k for example x_i. T is the number of weak classifiers in each subcategory classifier. In WTA-McBoost, we assume the highest score of all subcategories will be used to determine the fate of a given example. More specifically, let

$$y_i^T = \max_k y_{ik}^T. \tag{1}$$

Example x_i is classified as a positive example if y_i^T is greater than a threshold. Otherwise, it is a negative example. Following [12], we define the asymmetric boost loss as[1]:

$$L^T = \sum_{i=1}^N [I(z_i = 1) \exp\{-C_1 y_i^T\} + I(z_i = 0) \exp\{C_2 y_i^T\}], \tag{2}$$

where $C_1 > 0$ and $C_2 > 0$ are the cost factor of misclassification for positive and negative examples, respectively. $I(\cdot)$ is the indicator function. According to the statistical interpretation given by [4], minimizing this loss via boosting is equivalent to a stage-wise estimation procedure for fitting a cost-sensitive additive logistic regression model. In addition, as shown in [16], when $C_1 = C_2 = 1$, the above loss function is an upper bound of the training error on the data set $\mathcal{S}$.

Unfortunately, minimizing the loss function in Eq. (2) is difficult and can be very expensive in computation. Notice

$$\exp\{C_2 y_i^T\} = \exp\{C_2 \max_k y_{ik}^T\} \le \sum_k \exp\{C_2 y_{ik}^T\}, \tag{3}$$

we instead optimize a looser bound as:

$$\begin{aligned} L^T &= \sum_{i=1}^N \Big[I(z_i = 1) \exp\{-C_1 y_i^T\} + \\ &\quad I(z_i = 0) \sum_k \exp\{C_2 y_{ik}^T\} \Big]. \end{aligned} \tag{4}$$

Since the subcategories of the positive examples are different from each other, it is unlikely that a negative example having a high score in one subcategory will have high score in another category. Hence the looser bound in Eq. (4) shall still be reasonably tight.

In the following, we devise a two-stage algorithm to minimize the asymmetric boost loss in Eq. (4). With weak classifiers at stage t, define the current *run-time label* of positive example x_i as:

$$l_i^t = \arg\max_k y_{ik}^t. \tag{5}$$

Based on these labels, we can split the loss function into K terms, $L^t = \sum_{k=1}^K L_k^t$, where

$$\begin{aligned} L_k^t &= \sum_{i=1}^N \Big[I(l_i^t = k) I(z_i = 1) \exp\{-C_1 y_{ik}^t\} + \\ &\quad I(z_i = 0) \exp\{C_2 y_{ik}^t\} \Big], \end{aligned} \tag{6}$$

In the first stage of the algorithm, we assume the run-time labels are fixed, and search for the best weak classifiers $h_k^{t+1}(\cdot)$ and votes λ_k^{t+1} that minimize $\tilde{L}^{t+1} = \sum_{k=1}^K \tilde{L}_k^{t+1}$, where

$$\begin{aligned} \tilde{L}_k^{t+1} &= \sum_{i=1}^N \big[I(l_i^t = k) I(z_i = 1) \exp\{-C_1 y_{ik}^{t+1}\} + \\ &\quad I(z_i = 0) \exp\{C_2 y_{ik}^{t+1}\} \big]. \end{aligned} \tag{7}$$

This stage can be accomplished by performing boosting feature selection and vote computation for each subcategory *independently*. For instance, one can adopt the MBHBoost scheme proposed in [11], which trained multiple classes simultaneously and shared features among multiple classifiers. Alternatively, Appendix A presents a confidence-rated asymmetric boosting algorithm for the same purpose. Since the asymmetric boost loss is convex [12], it is guaranteed that this boosting step will reduce the loss function, i.e., $\tilde{L}_k^{t+1} \le L_k^t$, and $\tilde{L}^{t+1} \le L^t$.

In the second stage, we update the run-time labels, namely:

$$l_i^{t+1} = \arg\max_k y_{ik}^{t+1}. \tag{8}$$

The loss function is updated as $L^{t+1} = \sum_{k=1}^K L_k^{t+1}$, where

$$\begin{aligned} L_k^{t+1} &= \sum_{i=1}^N \big[I(l_i^{t+1} = k) I(z_i = 1) \exp\{-C_1 y_{ik}^{t+1}\} + \\ &\quad I(z_i = 0) \exp\{C_2 y_{ik}^{t+1}\} \big]. \end{aligned} \tag{9}$$

[1] We use asymmetric cost factors on positive and negative examples for the generalizability of the derivation. In our experiments on multiview face detection, we will only use $C_1 = C_2 = 1$, which produces satisfactory results. Asymmetric boosting could be very useful for other tasks such as learning a detector with very high detection rates [12].

Input

- Training examples $\{(x_i, z_i, s_i), i = 1, \cdots, N\}$, where $z_i \in \{0, 1\}$ for negative and positive examples. $s_i \in \{1, \cdots, K\}$ is the initial subcategory ID. For positive examples, s_i can be manually assigned or obtained through clustering. For negative examples, s_i can be randomly assigned (it will be ignored in WTA-McBoost).
- T is the total number of weak classifiers, which can be set through cross-validation.
- P is the label switching frequency.

Initialize

- Assign initial scores for all examples $y_{ik}^0 = 0$, where $k = 1, \cdots, K$ is the index of the subcategory classifier.
- Assign run-time label $l_i^0 = s_i$.

WTA-McBoost Learning

For $t = 1, \cdots, T$:

- According to the current run-time labels, perform confidence-rated asymmetric boosting for each subcategory (Eq. (7) and Appendix A).
- If $mod(t, P) = 0$, perform label switching for the positive examples ($z_i = 1$) according to Eq. (8).

Output

A set of K boosting classifiers $H_k^T(x_i) = \sum_{t=1}^{T} \lambda_k^t h_k^t(x_i)$.

Figure 1. WTA-McBoost learning.

It is straightforward to see that $L^{t+1} \leq \tilde{L}^{t+1}$, hence both stages of the algorithm will reduce the loss function. Given that the asymmetric boost loss in Eq. (4) is non-negative, the algorithm is guaranteed to converge to a (local) minimum.

The run-time labels in WTA-McBoost can be updated after each weak classifier is added. In practice, it may be beneficial to update them less frequently to avoid label oscillation. Fig. 1 shows the detailed stages of WTA-McBoost. In this flowchart the run-time-labels are updated every P weak classifiers are learned for each subcategory. A typical value of P is 32. Moreover, we may choose a large P at the very first round. This allows the subcategory classifiers to have a "burn in" period where they learn the general property of each subcategory. In our implementation we start label switching after 96 weak classifiers are learned for each subcategory. Since label switching involves little computation and it is done infrequently, the additional computational cost of WTA-McBoost compared with the traditional approach of training each sub-class separately is negligible.

Updating the run-time labels allows the positive example clusters to be re-formed during training, which can improve the final classification performance. In contrast, although clustering was used in [24] during sample set splitting, their clusters are fixed and do not change during feature selection. This may hinder the learning process due to misclassification during clustering. On the other hand, note in this paper we do not discuss subcategory splitting or merging. We may use manual labels to assign the initial subcategories, or we may use the splitting criteria and clustering method in [24]. In either case, WTA-McBoost can be applied to combat the misclassification issue and improve the overall classification performance.

3. Related Works and Discussions

Multiple category learning is closely related to multiple instance learning (MIL), e.g., [22]. In MIL, a bag of examples is classified as positive as long as one of the examples is positive. In MCL, an example is classified as positive as long as one of the subcategory classifiers classify it as positive. In both schemes there is uncertainty about which example/which subcategory shall be the best choice among a few. On the other hand, MIL and MCL have very different applications. MIL is often used to identify common objects in bags that have uncertainty in location or scale. MCL is more suitable for complex learning problems where it is desirable to cluster examples into different subcategories in order to improve the learning efficiency. Furthermore, in MIL a *single* classifier is learned throughout the process, while in MCL *multiple* classifiers are learned jointly.

The extension of MIL to MCL has been independently proposed by Kim and Cipolla [9] and Babenko et al. [1]. In both approaches, the formulation of MilBoost [22] was slightly extended. The training examples no longer have a fixed subcategory label. A set of likelihood values were maintained for each example, which describe the probability of it belonging to the subcategories during training. These likelihood values are combined with the same "noisy OR" scheme in [22] to compute the probability of the example being a positive example. To optimize the joint probability of all the training examples, AnyBoost [13] was adopted to derive the feature selection and vote computation during training, which requires a line search for each feature being tested. While such an extension shares similar ideas with the proposed WTA-McBoost, there is a significant difference when they are applied in real-world applications. Performing a line search for each feature candidate is a very computationally expensive operation, which is suitable for only small training sets and small number of features. The computational cost will increase further if feature sharing [20] shall be deployed, making the previous methods more unattractive. In contrast, as we have shown in Section 2, WTA-McBoost can be computed very efficiently, even with feature sharing.

In multiclass learning [16, 20], each example may have more than two labels or classes. The goal is to predict the examples' labels accurately by learning a single or multiple classifiers. Take face detection as an example. There are frontal faces, half-profile faces, profile faces, faces with 0 degree in-plane rotation, faces with ± 30 degree rotation, etc. Traditionally, faces are first labeled as one of the categories, and the detector is expected to output not only a

face/nonface decision, but also the pose of the face. It is therefore challenging to clearly define the goal of the learning algorithm – is the face/nonface decision more important, or the pose estimation more important? In contrast, in multiple category learning, we only focus on the face/nonface decision. By ignoring pose estimation, multiple category learning can make the face/nonface decision better due to adaptive labeling. One can always resort to a different classifier/regressor to estimate the pose thereafter.

Multitask learning [2, 20] is an approach to inductive transfer that improves learning for one task by using the information contained in the training signals of other related tasks. For instance, Torralba et al. [20] proposed to share features among multiple classifiers as a way to transfer information between them. Through information sharing, multitask learning improves generalization performance. Such an idea can be easily integrated to multiple category learning. In Section 4, we share features among multiple subcategory classifiers to build a multi-view face detector.

Recently, Dollar et al. [3] proposed multiple component learning. The algorithm learns component classifiers in a weakly supervised manner through multiple instance learning, where object labels are provided but part labels are not. It then uses these component classifiers as weak classifiers for a final boosting based strong classifier. Multiple component learning works well for articulated objects and is robust to occlusion. In essence, it can be viewed as an interesting way to create efficient weak classifiers. We may also combine it with multiple category learning by selecting weak classifiers among those prepared by multiple component learning.

Finally, we point out that when the total number of subcategories in multiple category learning is $K = 1$, WTA-McBoost will reduce to regular asymmetric boost [12]. In Appendix A we provide a confidence-rated prediction method for asymmetric boosting, which is very efficient to compute. In that algorithm, when the cost factors $C_1 = C_2 = 1$, asymmetric boosting reduces to regular AdaBoost and our confidence-rated prediction method reduces to the solution given in [16].

4. Experimental Results

4.1. Straightforward Application of WTA-McBoost

We first test the WTA-McBoost algorithm on a multi-view face detection problem in its most straightforward fashion. A total of 100,000 face images with size 24×24 pixels are collected from various sources including the web, the Feret database [14], the BioID database [7], the PIE database [19], etc. These faces are manually labeled into 5 subcategories: frontal, left half profile, left profile, right half profile and right profile. Each subcategory contains 20,000 faces, with -10 degree to +10 degree in-plane rotation. The negative image set is also collected from the web, which contains about 1.2 billion image patches of size 24×24 pixels.

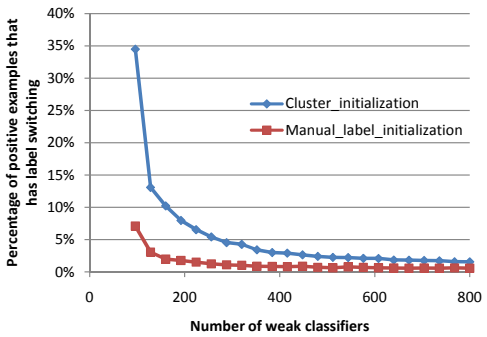


Figure 2. Percentage of positive examples that switch labels during WTA-McBoost.

We ran a simple k-means clustering algorithm on the face images, where the number of clusters is given as 5. The distance between two facial images is measured as the Euclidian distance between their down-sampled images (12×12 pixels). The initial means of the clusters are computed by averaging the images with the same manual labels. The k-means clustering algorithm converges in about 50 iterations.

Our experiment compares the learning performance of WTA-McBoost, and that of the traditional approach, which trains each subcategory separately. In fact, if we skip the label switching step in WTA-McBoost (Eq. (8)), we have an implementation of the traditional approach. In both cases, a total of 800 weak classifiers are learned for each category, with *shared* Haar features and shared feature partitions (see [21] for Haar features and Appendix A for the feature selection algorithm). Note feature sharing may speed up the detection speed and improve generalizability, but it is not required by either approach.

We show the percentage of positive examples that switch labels during WTA-McBoost in Fig. 2. Note as mentioned earlier, label switching starts at 96 weak classifiers, and is done once every 32 weak classifiers afterwards. It can be seen that during the first few rounds, many examples switched labels. In particular, at 96 weak classifiers, as many as 34.47% of the positive examples have switched labels for clustering based initialization, and 7.10% for manual labels based initialization. The number drops very quickly. At 480 weak classifiers, only 2.43% positive examples switched labels for clustering based initialization, and 0.72% for manual labels based initialization. This quick convergence of run-time labels can be very useful. For instance, once there are very few positive examples that will switch labels, a test example at this stage can be safely classified into one of the subcategories, and a single subcategory classifier can be run afterwards, which saves a lot of

Manual label	F	F	F	F	LHP	LHP	F	F	LHP	RHP
After WTA-McBoost	LP	LP	RP	RP	RP	RP	RHP	LHP	LP	LP
Manual label	F	F	LP	LP	RP	RP	LHP	LHP	RHP	RHP
After WTA-McBoost	LHP	LHP	LHP	LHP	RHP	RHP	LP	LP	RP	RP
Manual label	RP	LP	RP	RHP	RP	LHP	F	RHP	F	LHP
After WTA-McBoost	F	F	LHP	LHP	RHP	RHP	LP	LP	RP	RP

Figure 3. Training examples that switch their labels after WTA-McBoost. F: Frontal; LHP: Left Half Profile; RHP: Right Half Profile; LP: Left Profile; RP: Right Profile.

computation (see Section 4.2 for more details).

It is interesting to examine the training examples that switch their labels after WTA-McBoost. Fig. 3 shows a few such examples when the subcategory labels are initialized manually. In the first row, the examples all have very extreme lighting conditions. Such examples are abundant since we included the PIE database [19] in our training. We found that many of these examples have switched their labels after WTA-McBoost. The new labels are consistent in that when the lights are from the left, the examples tend to be relabeled as left profile, and when the lights are from the right, the examples tend to be relabeled as right profile. It appears that for these examples with extreme lighting, categorizing them to one of the profile subcategories help improve their detection accuracy. In the second row, we show some examples where the new labels are different from the manual label but very close. Such examples are also plenty. These examples show the unclear boundary between neighboring subcategories, and it is often hard for human to be certain which subcategory the examples shall be assigned. The third row shows a few examples where the new labels after WTA-McBoost do not seem to make much sense. Lucky, there are less than 50 such examples in the total set of 100,000 face images.

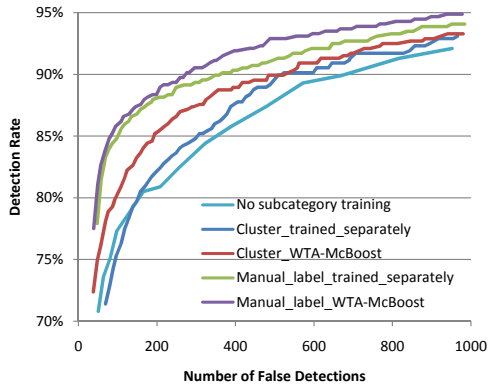
Finally, we test the learned detectors on two standard data sets that are never seen during training, the CMU+MIT frontal data set [15], and the CMU profile data set [17]. It is worth mentioning that the latter data set contains many faces that have more than ± 10 degree in-plane rotation, which are not represented in our training examples and *not* excluded in our experiments.

Fig. 4 shows the detector performance on the above two standard data sets. We also include a detector that is trained without subcategories with the learning framework in [26], i.e., faces of all poses are mixed together and a single boosting classifier is trained for face/non-face classification. The single boosting classifier uses the same Haar features and contains 2048 weak classifiers. A few observations can be made from Fig. 4. First, "divide and conquer" does help improve the performance. Even with the very naïve clustering based initialization, and all subcategories are trained separately, "divide and conquer" still outperforms the single boosting classifier trained with all poses mixed. Second, WTA-McBoost can improve the detector performance significantly compared with the traditional approach, even with manual labels as initialization. For instance, on the CMU+MIT frontal data set, at 85% detection rate, WTA-McBoost reduces the number of false detections by 37% for clustering based initialization; at 90% detection rate, WTA-McBoost reduces the number of false detections by 25% for manual label based initialization. Moreover, as mentioned earlier, WTA-McBoost requires negligible additional computation cost over the traditional approach, hence we recommend that WTA-McBoost shall always be used for training "divide and conquer" style multi-view object detectors.

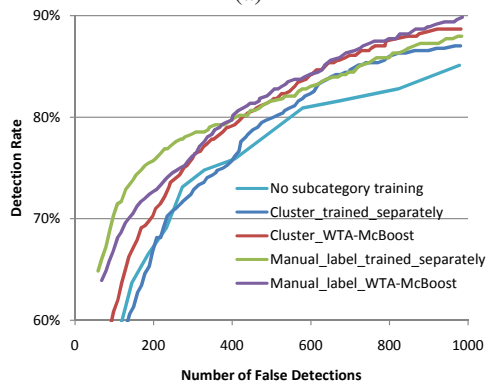
Another interesting observation from Fig. 4 is that detectors trained with manual label based initialization generally outperforms the naïve clustering based initialization. The WTA-McBoost algorithm is a greedy adaptive labeling algorithm. Similar to other greedy searching algorithms such as k-means clustering, the performance of the trained detector can vary given different initial labels, and good initial labels are always helpful in getting a good classifier. In practice, the initial labels are often given manually or automatically through clustering, in which case WTA-McBoost will almost always guarantee to derive a better classifier than the traditional approach of training each subcategory separately.

4.2. A Practical Multi-view Face Detector

Although features can be shared among the multiple classifiers learned with WTA-McBoost, the computational cost for classifying a test window is still relatively high for real-world applications. For instance, the time spent on



(a)



(b)

Figure 4. (a) Performance on CMU+MIT frontal data set (125 images, 483 labeled frontal faces). (b) Performance on CMU profile data set (208 images, 441 labeled faces with various poses, 73 (16.6%) of the faces have more than ±10 degree in plane rotations).

running a 5-category WTA-McBoost classifier with feature sharing is about 3 times that on a single category classifier. To improve the running speed, we propose to adopt a three-layer architecture for multi-view face detection, as shown in Fig. 5. More specifically, a single category classifier is first trained, which includes faces at all different poses. Although according to Fig. 4 a single category classifier trained with all poses may perform sub-optimally, this layer is critical in improving the detection speed. The second layer of the classifier is trained with WTA-McBoost, which allows the training positive examples to switch their subcategory labels during learning. As shown in Fig. 2, after a certain number of weak classifiers, the positive example clusters will converge, and the percentage of positive examples that switch labels during WTA-McBoost will be close to none. Once such a state has been achieved, we stop the WTA-McBoost learning, and train separate single category classifiers for each cluster.

During testing, a test window is first passed into the first layer single category classifier. With less than 100 weak classifiers, the classifier is often capable of removing about 90% of the negative windows. If the test window is not

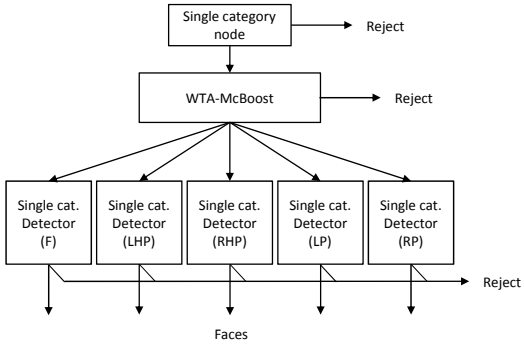


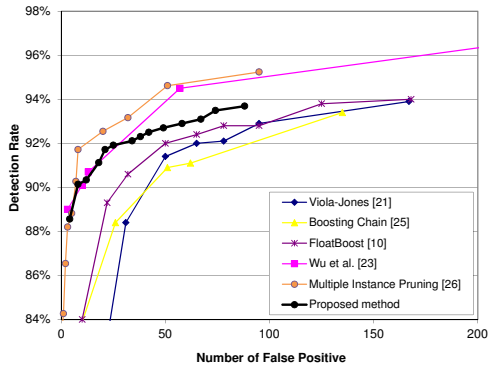
Figure 5. The three-layer architecture for multi-view face detection. Note in the third layer, the cluster IDs such as F, LHP, RHP, LP and RP are the output of WTA-McBoost and may not represent the true pose of the test window.

rejected by the first layer, it will be passed into the WTA-McBoost classifier. The second layer may reject another 80% of the remaining negative training examples. At the end of WTA-McBoost, the test window will be given a cluster ID based on the highest score of the multiple subcategory classifiers, which is then used to determine which third layer classifier will be run. This branching is safe because WTA-McBoost has already converged at this stage.

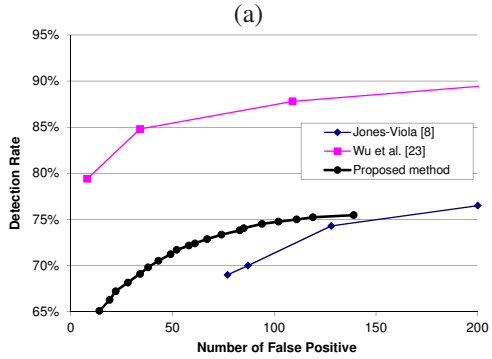
We trained a multi-view face detector with the above architecture. The positive examples were the same as those used in Section 4.1, and the negative example set was expanded to about 40 billion image patches. The first layer contains 64 weak classifiers, the WTA-McBoost based classifier contains 96 weak classifiers[2], and the third layer classifiers contains 608 weak classifiers each. Thanks to early rejection, the average number of weak classifiers visited per test window is 22, which is about 20% more compared with state-of-the-art fast frontal face detectors such as that in [26]. The running speed is also only about 20% slower than a single category face detector.

Fig. 6 compares the performance of our detector with a few existing approaches in the literature. It can been seen that on the CMU+MIT frontal data set, our detector's performance is comparable to many state-of-the-art algorithms. On the CMU profile data set, not many results have been reported in the past. The curve of Jones-Viola [8] was on profile face only (355 faces). Our detector's performance is noticeably lower than Wu et al. [23]. This can be attributed to two main reasons. First, about 16.6% of the faces in the CMU profile data set have more than ±10 degree in-plane rotation, which are not represented in our training examples and not excluded in our experiments. Second, we used the same Haar feature sets as [21] for training. Extending this features set may lead to much better performance for profile

[2]We used 32 weak classifiers as the burn-in period for WTA-McBoost, and switched labels every 8 weak classifiers. These settings are shorter than the examples we had in Section 4.1, but still worked fine. The short burn-in period is mostly due to speed concerns.



(a)



(b)

Figure 6. (a) Performance of our detector on CMU+MIT frontal data set (125 images, 483 labeled frontal faces). (b) Performance on CMU profile data set (208 images, 441 labeled faces with various poses, 73 (16.6%) of the faces have more than ± 10 degree in plane rotations).

face detection, as was reported in [25, 6].

5. Conclusions and Future Work

We have presented a winner-take-all multiple category boosting algorithm for learning complex object detectors through training multiple subcategory classifiers jointly. The key idea is to treat the initial subcategory labels as uncertain labels, and allow them to change during the training process. Through this process, positive examples that perform similarly measured by the learned subcategory classifiers will be organized together, which reduces the number of outliers in each subcategory and thus improves the overall learning performance.

We have showed in Section 4 the application of WTA-McBoost in learning a multi-view face detector, and we believe the same technique can be applied to other types of objects. There are a number of directions for future work. For instance, for a generic object class where there is no clear boundary between subcategories, it remains unclear how one would determine the best number of subcategories for training. Another interesting problem is how to set early pruning thresholds for a classifier learned by WTA-McBoost. There can be two principles in pruning such a classifier: one can terminate a subcategory classifier because a threshold score is not met, or because another subcategory classifier has reported a much higher score.

Appendix A: Confidence-Rated Asymmetric Boosting

In this Appendix we extend the confidence-rated prediction algorithm for AdaBoost [16] to asymmetric boosting. The result is similar to the solution provided in [12] but more explicit to compute. We also show that such a method can be used for learning multiple classifiers simultaneously with *shared* weak classifiers.

Consider a set of training examples as $\mathcal{S} = \{(x_i, z_i), i = 1, \cdots, N\}$, where $z_i = 1$ for positive examples and $z_i = 0$ for negative examples. Let the score of example x_i be $y_i^T = H^T(x_i) = \sum_{t=1}^{T} \lambda^t h^t(x_i)$, where T is the number of weak classifiers. For now let us assume a single category classifier is learned. The asymmetric loss function is:

$$L^T = \sum_{i=1}^{N} [I(z_i = 1) \exp\{-C_1 y_i^T\} + I(z_i = 0) \exp\{C_2 y_i^T\}]. \tag{10}$$

Given t weak classifiers selected, a new feature f^{t+1} and its J partitions $u_1, u_2, \cdots, u_J$, we first accumulate the weighted fraction of examples in each partition:

$$\begin{aligned} W_{+j} &= \sum_i I(f^{t+1}(x_i) \in u_j) I(z_i = 1) \exp\{-C_1 y_i^t\} \\ W_{-j} &= \sum_i I(f^{t+1}(x_i) \in u_j) I(z_i = 0) \exp\{C_2 y_i^t\}. \end{aligned} \tag{11}$$

Let the vote in partition u_j be c_j. In confidence-rated prediction, the score is computed as $y_i^{t+1} = \sum_{\tau=1}^{t+1} h^\tau(x_i)$. We have $h^{t+1}(x_i) = c_j, \text{if} f^{t+1}(x_i) \in u_j$. The loss function of partion u_j at $t+1$ is:

$$L_j^{t+1} = W_{+j} \exp\{-C_1 c_j)\} + W_{-j} \exp\{C_2 c_j\}. \tag{12}$$

It is easy to verify that when

$$c_j = \frac{1}{C_1 + C_2} \ln\left(\frac{C_1 W_{+j}}{C_2 W_{-j}}\right), \tag{13}$$

L_j^{t+1} has its minimum value as

$$\begin{aligned} L_j^{t+1} &= \gamma W_{+j}^{\frac{C_2}{C_1+C_2}} W_{-j}^{\frac{C_1}{C_1+C_2}}, \\ \text{where } \gamma &= \left(\frac{C_2}{C_1}\right)^{\frac{C_1}{C_1+C_2}} + \left(\frac{C_1}{C_2}\right)^{\frac{C_2}{C_1+C_2}} \end{aligned} \tag{14}$$

In practice, we search through all possible features and partitions to find the weak classifier that minimizes

$$L^{t+1} = \sum_j L_j^{t+1} = \gamma \sum_j W_{+j}^{\frac{C_2}{C_1+C_2}} W_{-j}^{\frac{C_1}{C_1+C_2}}. \tag{15}$$

Such feature selection and the vote computation in Eq. (13) can be implemented much more efficiently than a line search.

Furthermore, we can extend the above method to multiple category boosting with shared features. Given a feature, each subcategory classifier can find its corresponding partition and votes in order to minimize the joint loss of all K subcategory classifiers:

$$L^{t+1} = \sum_{k=1}^{K} \sum_j L_{kj}^{t+1}, \tag{16}$$

where L_{kj}^{t+1} is the loss for classifier k, partition j, computed by Eq. (14). The partition of the feature can also be shared by all the subcategory classifiers. The best feature is the one that minimizes the joint loss of Eq. (16).

References

[1] B. Babenko, P. Dollár, Z. Tu, and S. Belongie. Simultaneous learning and alignment: Multi-instance and multi-pose learning. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, 2008. 3

[2] R. Caruana. *Multitask Learning*. PhD thesis, Carnegie Mellon University, 1997. 4

[3] P. Dollar, B. Babenko, S. Belongie, and Z. Tu. Multiple component learning for object detection. In *Proc. of ECCV*, 2008. 4

[4] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics, Stanford University, 1998. 2

[5] C. Huang, H. Ai, Y. Li, and S. Lao. Vector boosting for rotation invariant multi-view face detection. In *Proc. of ICCV*, 2005. 1

[6] C. Huang, H. Ai, Y. Li, and S. Lao. Learning sparse features in granular space for multi-view face detection. In *Intl. Conf. on Automatic Face and Gesture Recognition*, 2006. 7

[7] O. Jesorsky, K. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. *Audio and Video based Person Authentication - AVBPA 2001*, pages 90–95, 2001. 4

[8] M. Jones and P. Viola. Fast multi-view face detection. Technical report, Mitsubishi Electric Research Laboratories, TR2003-96, 2003. 1, 6

[9] T.-K. Kim and R. Cipolla. MCBoost: Multiple classifier boosting for perceptual co-clustering of images and visual features. In *Proc. of NIPS*, 2008. 3

[10] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. In *Proc. of ECCV*, 2002. 1

[11] Y.-Y. Lin and T.-L. Liu. Robust face detection with multi-class boosting. In *Proc. of CVPR*, 2005. 2

[12] H. Masnadi-Shirazi and N. Vasconcelos. Asymmetric boosting. In *Proc. of ICML*, 2007. 2, 4, 7

[13] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Proc. of NIPS*, 2000. 3

[14] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi. The feret evaluation methodology for face recognition algorithms. *IEEE Trans. on PAMI*, 22(10):1090–1104, 2000. 4

[15] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. on PAMI*, 20:23–38, 1998. 5

[16] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999. 1, 2, 3, 4, 7

[17] H. Schneiderman and T. Kanade. A statistical model for 3d object detection applied to faces and cars. In *Proc. of CVPR*, 2000. 5

[18] E. Seemann, B. Leibe, and B. Schiele. Multi-aspect detection of articulated objects. In *Proc. of CVPR*, 2006. 1

[19] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Trans. on PAMI*, 25(12):1615–1618, 2003. 4, 5

[20] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. on PAMI*, 29(5):854–869, 2007. 3, 4

[21] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of CVPR*, 2001. 1, 4, 6

[22] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *Proc. of NIPS*, volume 18, 2005. 3

[23] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. In *Proc. of IEEE Automatic Face and Gesture Recognition*, 2004. 1, 6

[24] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *Proc. of ICCV*, 2007. 1, 3

[25] R. Xiao, L. Zhu, and H. Zhang. Boosting chain learning for object detection. In *Proc. of ICCV*, 2003. 7

[26] C. Zhang and P. Viola. Multiple-instance pruning for learning efficient cascade detectors. In *Proc. of NIPS*, volume 20, 2007. 5, 6