Efficient Incremental Learning of Boosted Classifiers for Object Detection

Pramod Sharma[1], Chang Huang[2*] and Ram Nevatia[1]
[1]*University of Southern California, Los Angeles, CA*
[2]*NEC Labs, Cupertino, CA*
{*pksharma,huangcha,nevatia*}*@usc.edu*

Abstract

Significant progress has been made towards learning a generalized offline object detector. However, when a generalized offline detector is applied on new datasets, it often makes mistakes by missing some specific instances of the object or by producing false detections in the background scene. In order to rectify these mistakes made by the offline detector, we present a novel and efficient incremental learning method , which adjusts the parameters of offline trained cascade of boosted classifiers using manually labeled online samples. Experiments demonstrate both the efficiency and effectiveness of our approach.

1. Introduction

In many object detection methods a generalized offline detector is trained by collecting thousands of positive and negative training examples on the assumption that these examples would represent the objects present in the unseen test data. Offline detector trained in this manner, may not work for some specific cases. One possible solution to handle these special cases is to repeat the offline training process by including the special cases in the training set. However re-training for every new dataset is expensive.

We present a novel and efficient incremental learning method which addresses these issues. Our method finds the optimal adjustments to the parameters of offline learned detector by optimizing a hybrid loss function, which is a combination of offline and online loss functions. We estimate the loss incurred by offline samples, without using offline samples during incremental learning, and combine this offline loss with loss incurred by online samples. Hybrid loss function is optimized for one parameter at a time and an exact solution is found for the loss function optimization problem, which makes our method computationally efficient. Overview of our approach is shown in Figure 1.

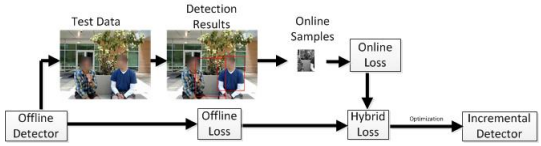


Figure 1. Overview of our approach

In recent years many approaches have been proposed for incremental and online learning [1, 3, 10, 7, 9, 5, 8]. Main focus has been on the classifiers based on SVM [9, 8] and boosting [1, 3, 10]. Kembhavi et al.[9] introduced incremental multiple kernel learning for object recognition. Joshi and Porikli [8] used SVM for incremental active learning. However, time can be a critical constraint for incremental learning and SVM based methods are in general computationally expensive. Huang et al. [5] proposed an incremental learning method for Real Adaboost framework. In [5], loss functions are defined as a function of all the parameters of the offline learned classifier and is optimized using steepest descent method, which is inherently slow in nature because of its convergence time.

2 Incremental Learning

Real Adaboost learns weak hypotheses $\{h_1, h_2, ..., h_t\}$ and combination coefficients $\boldsymbol{\theta} = \{(\theta_{11}, .., \theta_{1m_1}), .., (\theta_{i1}, .., \theta_{im_i}), .., (\theta_{T1}, ., \theta_{Tm_T})\}$, where m_i is the total number of partitions for i^{th} weak hypothesis. These weak hypotheses are selected by using thousands of training samples during offline training, which is important in order to learn discriminative weak hypotheses which can differentiate one category from the other.

However, for incremental learning, only few online examples may be available, which may not be sufficient enough to add/remove additional weak hypothe-

*This work was done when Chang Huang was with the University of Southern California, Los Angeles, CA.

ses, hence in our incremental learning method, we do not modify the weak hypotheses, instead we focus on finding the optimal addition ($\boldsymbol{\Delta\theta}$) to the offline learned combination co-efficients ($\boldsymbol{\theta}$) for these weak hypotheses.

We initialize each element in $\boldsymbol{\Delta\theta}$ equal to zero and iterate over all the weak hypotheses of the strong classifier sequentially for the optimal adjustment to $\boldsymbol{\Delta\theta}$ and update each element in $\boldsymbol{\Delta\theta}$ after optimization. If θ_{ij} is combination co-efficient of j^{th} partition of the i^{th} weak hypothesis. we find $\Delta\theta_{ij}^{'}$ by minimizing the hybrid loss function $L(P(x,y) : \boldsymbol{\theta}^{'} + \Delta\theta_{ij}^{'})$, which is combination of offline loss function $\bar{L}(P_{off}((x|y), \boldsymbol{\theta}^{'} + \Delta\theta_{ij}^{'})$ and online loss function $L(P_{on}(x \mid y), \boldsymbol{\theta}^{'} + \Delta\theta_{ij}^{'})$ and is defined as:

$$L(P(x,y) : \boldsymbol{\theta}^{''}) = \alpha_y \cdot P(y) \cdot L(P_{on}^{(x|y)}, \boldsymbol{\theta}^{''}) + (1-\alpha_y) \cdot P(y) \cdot \bar{L}(P_{off}^{(x|y)}, \boldsymbol{\theta}^{''}) \quad (1)$$

where $\boldsymbol{\theta}^{''} = \boldsymbol{\theta} + \boldsymbol{\Delta\theta} + \Delta\theta_{ij}^{'}$. $P_{off}^{(x|y)}$, $P_{on}^{(x|y)}$ are offline and online likelihood respectively.$P(y)$ is prior probability for category y and $y \in \{-1, +1\}$. α_y is a regularization parameter which decides the weights given to the offline and online part during incremental learning. After optimization, we update $\Delta\theta_{ij}$ in $\boldsymbol{\Delta\theta}$ as:

$$\Delta\theta_{ij} = \Delta\theta_{ij} + \Delta\theta_{ij} \quad (2)$$

In following subsections, we describe the estimation of offline and online loss functions and optimization solution of hybrid loss function. Then we discuss the time complexity of our method.

2.1 Offline Loss Estimation:

In [5], estimated offline loss $\bar{L}(P_{off}^{(x|y)}, \boldsymbol{\theta} + \boldsymbol{\Delta\theta})$ is defined as:

$$\bar{L}(P_{off}^{(x|y)}, \boldsymbol{\theta} + \boldsymbol{\Delta\theta}) = L(P_{off}^{(x|y)}, \boldsymbol{\theta}) \cdot \prod_i \sum_j \widehat{P}_{off}(z_{i,j} \mid y) \cdot \exp(-y\Delta\theta_{ij}) \quad (3)$$

where $\widehat{P}_{off}(z_{i,j} \mid y)$ is the weighted marginal likelihood for j^{th} partition of i^{th} weak hypothesis, More detailed description about the derivation of Eqn. 3 can be found in [5].

By using Eqn. 3, we can define our offline loss function as:

$$\bar{L}(P_{off}^{(x|y)}, \boldsymbol{\theta}^{'} + \Delta\theta_{ij}^{'}) = \bar{L}(P_{off}^{(x|y)}, \boldsymbol{\theta}) \prod_{t \in 1 \ldots T/i} \sum_m \widehat{P}_{off}(z_{t,m} \mid y) \exp(-y\Delta\theta_{tm}) \left(\left(\sum_{m/j} \widehat{P}_{off}(z_{i,m} \mid y) \exp(-y\Delta\theta_{im}) \right) + \widehat{P}_{off}(z_{i,j} \mid y) \exp(-y\Delta\theta_{ij}) \exp(-y\Delta\theta_{ij}^{'}) \right) \quad (4)$$

2.2 Online Loss Estimation:

For online samples, total online loss is estimated as:

$$L(P^{on}) = \sum_{k=1}^{N_{on}^{y}} \frac{1}{N_{on}^{y}} \exp(-yG(F(x_k) : \boldsymbol{\theta}^{'})) \quad (5)$$

where $P^{on} = (P_{on}(x \mid y), \boldsymbol{\theta}^{'})$, $\boldsymbol{\theta}^{'} = \boldsymbol{\theta} + \boldsymbol{\Delta\theta}$, N_{on}^{y} is total number of online samples for category y, and

$$G(F(x_k) : \boldsymbol{\theta}^{'}) = \sum_{m=1}^{T} g_m(f_m(x_k) : \boldsymbol{\theta}^{'}) \quad (6)$$

where T is total number of weak hypotheses.

While finding optimal adjustment for θ_{ij}, we can define our online loss function as:

$$L(P_{on}(x \mid y), \boldsymbol{\theta}^{'} + \Delta\theta_{ij}^{'}) = \sum_{k=1}^{N_{on}^{y}} \frac{[f_i(x_k)]}{N_{on}^{y}} \exp(-yG(F(x_k) : \boldsymbol{\theta}^{'})) \exp(-y\Delta\theta_{ij}^{'}) \quad (7)$$

where $[f_i(x_k)] = 1$, if $f_i(x_k) = j$, 0, otherwise.

2.3 Solution for adjustment of combination coefficients:

We define Ω_y^{off} and Ω_y^{on} as follows:

$$\Omega_y^{off} = -y\bar{L}(P_{off}^{(x|y)}, \boldsymbol{\theta})\widehat{P}_{off}(z_{i,j} \mid y) exp(-y\Delta\theta_{ij}) \prod_{t \in 1 \ldots T/i} \sum_m \widehat{P}_{off}(z_{t,m} \mid y) exp(-y\Delta\theta_{tm}) \quad (8)$$

$$\Omega_y^{on} = \alpha_y^{'} \sum_{k=1}^{N_{on}^{y}} \frac{[f_i(x_k)]}{N_{on}^{y}} exp(-yG(F(x_k) : \boldsymbol{\theta}^{'})) \quad (9)$$

where $\alpha_y^{'} = \alpha_y P(y)$. By differentiating the offline loss function (Eqn. 4) w.r.t. $\Delta\theta_{ij}^{'}$ and using Ω_y^{off} (Eqn. 8) :

$$(1-\alpha_y)P(y)\frac{\partial \bar{L}(P_{off}^{(x|y)}, \boldsymbol{\theta}^{'} + \Delta\theta_{ij}^{'})}{\partial \Delta\theta_{ij}^{'}} = -\Omega_{+1}^{off} exp(-\Delta\theta_{ij}^{'}) + \Omega_{-1}^{off} exp(\Delta\theta_{ij}^{'}) \quad (10)$$

Similarly, by differentiating the online loss function (Eqn. 7) w.r.t. $\Delta\theta_{ij}^{'}$ and by using Ω_y^{on} (Eqn. 9), we can write:

$$\alpha_y P(y)\frac{\partial L(P_{on}(x|y), \boldsymbol{\theta}^{'} + \Delta\theta_{ij}^{'})}{\partial \Delta\theta_{ij}^{'}} = -\Omega_{+1}^{on} exp(-\Delta\theta_{ij}^{'}) + \Omega_{-1}^{on} exp(\Delta\theta_{ij}^{'}) \quad (11)$$

Now if we solve for $\frac{\partial L(P(x|y), \boldsymbol{\theta}^{'} + \Delta\theta_{ij}^{'})}{\partial \Delta\theta_{ij}^{'}} = 0$, by using Eqn. 10 and Eqn. 11, we get $\Delta\theta_{ij}^{'}$ as:

$$\Delta\theta_{ij}^{'} = \frac{1}{2} \log \frac{(\Omega_{+1}^{off} + \Omega_{+1}^{on})}{(\Omega_{-1}^{off} + \Omega_{-1}^{on})} \quad (12)$$

The incremental learning process for a strong classifier is described in Algorithm 1.

2.4 Time Complexity

Time is a critical factor for incremental learning methods. If N^h are the number of weak hypotheses in the strong classifier, N^o are the number of online samples and N^p corresponds to the number of partitions in a hypothesis. Then time complexity for our algorithm would be $\mathcal{O}\left(N^h * (N^o * N^{p_h})\right)$.

For steepest descent methods, other than the cost of computing all the gradients which is $\mathcal{O}\left(N^h * (N^o * N^{p_h})\right)$, there is additional overhead of dealing with Hessian approximation at each iteration, which can be computationally expensive if the number of parameters to optimize are large. In cascade Real Adaboost classifier, a strong classifier can have hundreds of weak classifiers and each weak classifier can have tens of partitions, so the number of parameters to optimize can be in thousands. Therefore, for such a large parameter set, steepest descent methods can be computationally expensive.

Algorithm 1 Real Adaboost Incremental Learning for a strong classifier

- **Given**: Online Samples = $\{(x_1, y_1), ..., (x_n, y_n)\}$, Strong classifier $\mathbf{H} = \{h_1, h_2, ..., h_T\}$.
- **Init**: α_{+1}, α_{-1}; $\boldsymbol{\Delta\theta} = 0$.

for $i = 1$ to T **do**

- For each partition $j \in h_t$, compute
  compute Ω_{+1}^{off} and Ω_{-1}^{off}
  compute Ω_{+1}^{on} and Ω_{-1}^{on}
  $\Delta\theta_{ij}^{'} = \frac{1}{2}\log\{\frac{(\Omega_{+1}^{off}+\Omega_{+1}^{on})}{(\Omega_{-1}^{off}+\Omega_{-1}^{on})}\}$
  $\Delta\theta_{ij} = \Delta\theta_{ij} + \Delta\theta_{ij}^{'}$

Obtain h_i^* (with updated combination co-efficients)

Output strong classifier $\boldsymbol{H^*} = (h_1^*, h_2^*, ..., h_T^*)$

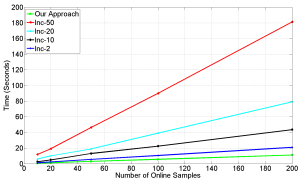
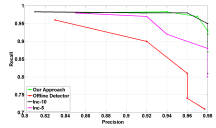
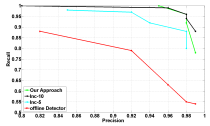


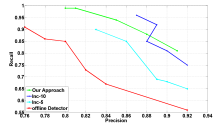
Figure 2. Comparison of the running time of our incremental learning approach with approach described in [5]. Inc-T (T=2,10, 20, 50) represents maximum T iterations of incremental learning using the method described in [5].



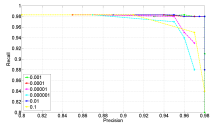
(a) ROC-M1



(b) ROC-M2



(c) ROC-D1



(d) Regularization Parameter

Figure 3. ROC curves and Regularization Parameter

3 Experiments

We evaluated our approach for the problem of pedestrian detection. Two different human datasets are used for evaluation: UCR dataset [2] and UCSD dataset [1]. First we compare the running time of our approach with the approach described in [5]. Then we present the detection performance on UCR and UCSD Dataset.

3.1 Time Computation Performance

For the comparison of computational time we train the upper body of the human detector offline by collecting around 20,000 training samples from the Internet. The offline detector is trained for 21 layers of cascade Real Adaboost using the method described in [6]. We run the experiments for our approach and [5], on a 3.16 GHz XEON CPU.

In Figure 2 we show the comparison of computational time of our method with [5] for different number of online samples used for incremental learning. We can notice that our approach takes around only 1 second for 10 online samples.

3.2 Detection Performance

Datasets used: To evaluate the detection performance of our incremental learning method, we use three sequences; two sequences from UCR dataset: "'M1:Two people meeting on a bench without gestures.avi"' (M1), "'M2:Two people meeting on a bench with gestures.avi"'(M2), and and David Indoor (D1) sequence from UCSD dataset. For the evaluation of UCR videos, we sample 94 frames from each video randomly from the latter half of the sequence and manually label the humans in these frames. For M1, 186 humans are labeled, whereas for M2, 182 humans are labeled as ground-truth. For D1 sequence, we sample total 80 frames from

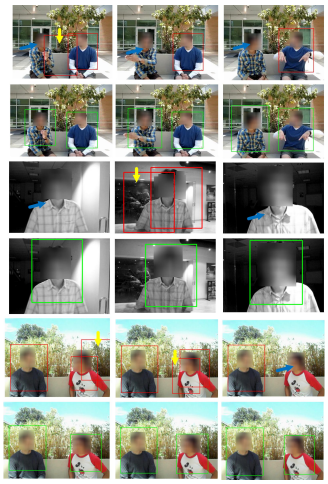


Figure 4. Detection Results from offline detector [6](odd rows, Red) and our approach (even rows, Green). Our approach is able to recover the missed detections (marked in blue) and false alarms (marked in yellow) produced by the offline detector.

the sequence and manually label 80 humans in these frames as ground-truth.

Online Sample collection: For each sequence, we annotate only 1 missed detection from the first half of the sequence and perturb this annotated example to generate 10 online positive samples and collect 10 false alarms from the background as the online negative samples.

Choice of regularization parameter: impact of changes in α_y on detection performance is shown in Figure 3(d). We use 10 online samples for this experiment, whereas offline samples used to train the offline detector (described in section 3.1) are in ten thousands, hence we get high performance if α_y is set (equal to 0.001) to close to the ratio of number of online samples to the number of offline samples. If we use very small $\alpha_y(10^{-6})$ or very high $\alpha_y(0.1)$, the performance deteriorates.

Experiment settings and evaluation criteria:: We use the offline detector trained for evaluating the computation time performance as described in section 3.1. regularization parameter (α_y)is set to 0.001 for all the experiments. We use the Recall-Precision criteria to evaluate the detection performance and follow the 50% overlap criteria as used in [6].

Detection Results:We compare detection results from our approach with offline detector and method used in [5]. From Figure 3, we can see that both our approach and steepest descent method described in [4] works better than the offline detector. Our approach gives better results than the 5 iterations of the steepest descent method for all the sequences, whereas results from our method are similar to the 10 iterations of the steepest descent method. Few examples of detection results are shown in Figure 4.

4 Conclusion

In this paper, we presented an efficient incremental learning method for cascade Real Adaboost classifier. We combine the offline and online loss to make a hybrid loss function and propose an efficient method to optimize this hybrid loss function. Our experiments on the problem of pedestrian detection demonstrate that our method improves the performance of an offline trained detector significantly by collecting only few online samples.

References

[1] B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009.

[2] G. Denina, B. Bhanu, H. T. Nguyen, C. Ding, A. Kamal, C. Ravishankar, A. Roy-Chowdhury, A. Ivers, and B. Varda. Videoweb dataset for multi-camera activities and non-verbal communication. In B. Bhanu, C. V. Ravishankar, A. K. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos, editors, *Distributed Video Sensor Networks*, pages 335–347. Springer London, 2011.

[3] H. Grabner and H. Bischof. On-line boosting and vision. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 260 – 267, 2006.

[4] C. Huang, H. Ai, B. Wu, and S. Lao. Boosting nested cascade detector for multi-view face detection. In *ICPR (2)*, pages 415–418, 2004.

[5] C. Huang, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Incremental learning of boosted face detector. In *ICCV*, pages 1–8, 2007.

[6] C. Huang and R. Nevatia. High performance object detection by collaborative learning of joint ranking of granules features. In *CVPR*, pages 41–48, 2010.

[7] O. Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *CVPR (1)*, pages 696–701, 2005.

[8] A. J. Joshi and F. Porikli. Scene-adaptive human detection with incremental active learning. *Pattern Recognition, International Conference on*, 0:2760–2763, 2010.

[9] A. Kembhavi, B. Siddiquie, R. Miezianko, S. McCloskey, and L. Davis.

[10] N. C. Oza and S. Russell. Online bagging and boosting. In *In Artificial Intelligence and Statistics 2001*, pages 105–112. Morgan Kaufmann, 2001.