Extensive Articulated Human Detection by Voting Cluster Boosted Tree

Bo Yang, Chang Huang and Ram Nevatia
University of Southern California, Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089, USA
{yangbo|huangcha|nevatia}@usc.edu

Abstract

Our goal is to detect people in highly articulated poses, including bending, crouching, etc. Such formidable diversity in human poses makes detection much more difficult than for pedestrian poses. "Divide-and-conquer" is a favorable strategy for detecting objects with large intra class variations, which splits object instances into several sub-categories and trains relatively simple classifiers for each sub-category. We propose a novel sample split method, which benefits the learning results of articulated humans. We adopt the Cluster Boosted Tree (CBT) structure to automatically decide when a split should be triggered. Unlike the simple k-means used in CBT for sample split, our approach aims at minimizing the training loss after the split. Since this minimization is an NP-hard problem, we design a heuristic algorithm, in which we find optimal sample divisions according to each single feature, and then make compromises to get a final division by a voting-like process. We name our training method as Voting Cluster Boosted Tree (VCBT). Furthermore, to avoid large background area in training samples, we first cluster samples according to their width/height ratios, and then train a VCBT for each subset. We conduct an experiment on 17 infrared surveillance video clips, report superior performance compared with previous human detection methods, and show how our approach benefits the learning results by reducing training loss.

1. Introduction

Human detection is one of the fundamental problems in computer vision area. Owing to its considerable impact on a wide range of applications, many researchers have put great efforts on this topic, and have made some significant progress [18][4][13][10][2][7]. However, almost all of them focus on “pedestrian detection”. As an articulated object, a person may be in multiple poses as shown in Figure 1, e.g., standing, bending, crouching and sitting, while typical pedestrian poses only include walking and standing.

Detection of humans with highly articulated poses is a much more difficult problem than detection of pedestrians. Even for pedestrians, the appearance may change with illumination, view point, small articulations. For highly articulated humans, their appearance may change much more, and therefore may have quite complicated distributions in the feature space. This indicates that training all samples together may have over-fit effects; in addition, finding templates for all samples is difficult. One reasonable approach is “divide-and-conquer”, i.e., to split samples into different clusters, so that samples in the same subset are similar, and training for each subset becomes practical. The sample split approach can be supervised [8] or unsupervised [17], visual appearance based [23] or feature based [20]. In our scenario, it is difficult to pre-define clusters, as articulated humans poses are diverse. Moreover, Wu and Nevatia [20] have shown that feature based split is superior to appearance based split in terms of detection accuracy, because the former uses the same features for clustering and detection. On the other hand, although sample split is necessary, the cluster number should not be large. As one cluster has one classifier and the input image will be scanned by all classifiers, their false alarms will sum together, which makes the performance more sensitive to noise compared with single classifier. Therefore, we adopt the unsupervised feature
based cluster boosted tree (CBT) [20] structure, because it splits samples with large variation automatically and only splits when necessary.

We argue that the sample split method used in CBT [20] (simple k-means) only considers the feature similarity of positive samples and does not necessarily benefit the boosting results. We introduce a voting process into CBT, named Voting-CBT (VCBT), which heuristically optimizes the sample split strategy in the boosting process by considering both positive and negative samples. In the boosting process, all positive samples are trained together at first, and weak classifiers are added continually. When new weak classifiers contribute little, a sample split is made and a re-training process starts. After that each subset is trained separately and does not share features with the others. As we cannot know what features would be selected after the split, we argue that what we can do to benefit the boosting process is just minimize the training loss, i.e., the upperbound of training errors, on existing features. Unfortunately, this problem is NP-hard, but finding solutions on one feature is affordable. Therefore, we design a heuristic algorithm to minimize the loss. We first get best divisions according to each single feature independently, and then all divisions vote for the final partition according to the weights of features, which are inversely proportional to training loss on each single feature.

We test our approach on 17 infrared surveillance video clips, in which people act naturally. The results indicate that our approach not only achieves lower training loss, but also has better performance on test data. Moreover, the final detection results are also improved. Although our experiments are done on infrared data, our approach is general and not restricted to specific features or data types.

The rest of the paper is organized as follows: Section 2 discusses the related work; Section 3 gives a rough review on cluster boosted tree (CBT), and describes our sample division algorithm; experimental results are shown in Section 4, followed by conclusions in Section 5.

2. Related Work

There is large amount of work on human detection, more specifically, pedestrian detection. Researchers have developed diverse features to describe pedestrians, such as Haar-like features [18], HOG [4], Shapelet [13], Edgelet [22]. Although some features may be pose-invariant for pedestrians [9], more highly articulated humans may have complicated distribution in these feature spaces. Part-based representation and joint analysis have shown excellent performance on occluded human detection [16] [9] [22], which can adapt to some pose changes. However, almost all of these methods are based on an assumption that head is above torso, and torso is above legs in the image. Such assumption is not always valid for our scenario, as shown in Figure 1.

To deal with detection of objects with large variation, researchers have developed many sample split methods. Huang et al [8] predefined categories by visual appearance and domain knowledge, and then trained a feature sharing cascade with several branches. Since domain knowledge is not always easy to get, based on human silhouettes, Seemann et al [15] and Zhang et al [23] used unsupervised clustering algorithm to automatically divide samples into different subsets, so that humans in similar poses tend to cluster together. However, as shown in [23], many clusters are required to represent humans in diverse poses. As these clusters are trained separately, there will be many detectors scanning the target image, and their false alarms will sum together. Even if each detector has reasonable false alarm rate, the final precision may be quite low. To avoid this problem, Wu and Nevatia [20] proposed the cluster boosted tree to automatically split samples with minimal clusters, and showed excellent performance on multi-view pedestrian detection.

Another related area is human pose estimation with rich literature [1][11][19][6]. These approaches estimate body parts, including head, torso, arms, legs, and further infer human poses or actions. Although these approaches do not make assumptions of human poses, most of them work on single person without any occlusion and require an estimate of human’s position as an input. In addition, pose estimation tasks often require much higher resolution than detection ones do.

3. Approach

In this section, we first review the standard real AdaBoost [14], which is the basis of CBT [20] and our VCBT. Then we will have a rough review of the CBT algorithm, and detail our sample division strategy.

3.1. AdaBoost Algorithm

Let a labeled training set be \( S = (x, y) \), where \( x \) is an image sample, and \( y = \pm 1 \) indicating \( x \) as positive or negative. A cascade classifier \( H \) is defined as

\[
H(x) = \sum_i h_i(f_i(x))
\]

(1)

where \( f_i \) is a feature which maps the image sample to a real value, and \( h_i \) is a weak classifier learnt from \( f_i(x) \). Let the value range of \( f_i(x) \) is \( R_i \). We divide \( R_i \) into \( n \) non-overlap equal sized continuous ranges, so that

\[
R_i = \bigcup_{j=1}^n F_{ij}, \quad F_{ij} \cap F_{ik} = \Phi \quad \forall j \neq k
\]

(2)
Then the weak classifier $h_j$ is defined as

$$ if \ f_i(x) \in F_{ij}, \ h_i(x) = \frac{1}{2} \ln \left( \frac{W^+_j + \epsilon}{W^-_j + \epsilon} \right), \ j = 1, \ldots, n $$

where $\epsilon$ is a smoothing factor, and $W^+_j (b \in \{1, -1\})$ is the probability distribution of $f_i(x)$ for all samples $x$. In practice, to avoid side effects coming from unbalanced sizes of positive and negative samples, we often normalize the weights of positive and negative samples separately. Thus, $W^+_j$ is defined as

$$ W^+_j = \sum_{f_i(x) \in F_{ij}, y=b} w(x), \ b \in \{1, -1\}, \ j = 1, \ldots, n $$

where $w(x)$ is the weight for sample $x$.

As there are often too many features and not all of them are discriminative, in each boosting round, the feature that best distinguishes positive samples from negative ones is selected, and the weak classifier based on it is added into the cascade classifier. According to the definition of $W^+_j$, the power of one weak classifier is defined as

$$ Z \approx \sum_j \sqrt{W^+_j W^-_j} $$

The smaller $Z$ is, the more powerful a weak classifier is.

### 3.2. Cluster Boosted Tree Algorithm

When there are large variations in positive sample set, the features may be not discriminative enough. After several boosting rounds, $Z$ values will grow up to almost 1. That means the classification task is beyond the ability of these features. In cluster boosted tree, if $Z$ values are larger than a threshold for several rounds, a split is triggered.

In CBT [20], a feature vector $f = (f_1, \ldots, f_K)$ is generated for each positive sample, in which $f_1, \ldots, f_K$ are selected features by the boosting process. Then a k-means is adopted on such vectors to divide positive samples into two clusters. Then, a retrain process starts. It does not change existing selected features, but only tunes the threshold for each subset. After retraining, the two subsets are continually trained by real AdaBoost algorithm separately, until classifiers for all subsets reach the target training errors.

### 3.3. Sample Division in VCBT

As discussed in Section 1, since we cannot know what features are selected after the split, we aim at minimizing the training loss on existing features. Schapire and Singer [14] had given the upper bound of training errors as $\prod_{t=1}^{T} Z_t$, in which $T$ is the number of weak classifiers, and $Z_t$ is defined as Equ.5. Therefore, when a split occurs, our target is to minimize the larger one of the two upper bounds of training errors. However, find a best division on $T$ weak classifiers is NP-hard. Thus, we first consider an easy case that $T = 1$, and then get the division for multiple weak classifiers.

#### 3.3.1 Division on One Feature

Let us suppose there is only one feature $f$ selected before the split. Let the probability distribution vector on feature $f$ as $(W^+_1, \ldots, W^+_n) (b \in \{1, -1\})$ for positive and negative samples respectively. Then, after the split, the negative distribution vector does not change, but the positive one will be divided into two as $(a^+_1, \ldots, a^+_n)$ and $(b_1^+, \ldots, b^n_+)$.

Thus, we have

$$ W^+_j = a^+_j + b^+_j, \ a^+_j \geq 0 \ and \ b^+_j \geq 0, \ j = 1, \ldots, n $$

Note that after the split, weights for samples in different subsets will be normalized separately by Equ.4. Therefore, let $A = \sum_j a^+_j$ and $B = \sum_j b^+_j$, and then the problem becomes solving the following formula

$$ \arg\min_{(a^+_1, \ldots, a^+_n)} (\max \left( \sum_j \sqrt{a^+_j W^+_j / A}, \sum_j \sqrt{b^+_j W^+_j / B} \right)) $$

As minimizing the larger value of two is difficult to solve, instead of Equ.7, we minimize the sum of the two. Let $g_j(a^+_j) = \sqrt{a^+_j W^+_j / A} + \sqrt{b^+_j W^+_j / B}$. Then the new minimization problem can be formulated as

$$ \arg\min_{(a^+_1, \ldots, a^+_n)} \left( \sum_j g_j(a^+_j) \right) $$

As one positive sample can only fall into one bin of the feature, minimizing $g_j(a^+_j)$ is independent with others. Thus, we minimize $g_j(a^+_j)$ one by one for all $j$. When minimizing $g_j$, we assume $A$ and $B$ as constants. This is because $a_j$ and $b_j$ will only occupy a small proportion to $A$ and $B$ when the bin number $n$ is large $^1$, and the change of $a_j$ and $b_j$ will not have much effect on $A$ and $B$. By such simplification, we get the derivative of $g_j$ as

$$ \frac{dg_j}{da^+_j} = \sqrt{\frac{W^+_j}{2}} \left( \frac{1}{a^+_j A} - \frac{1}{(W^+_j - a^+_j)B} \right) $$

Equ.9 shows that the derivative of $g_j$ decreases from $+\infty$ to $-\infty$ with the increase of $a^+_j$ from 0 to $W^+_j$, indicating that $g_j$ first increases and then decreases. Therefore, to minimize $g_j$, we have $a^+_j = 0$ or $W^+_j$. That means samples belong to the same bin should be divided into the same subset. However, according to this rule, there are still $2^{n-1}$ ways

---

$^1$We set $n=32$ in our experiment.
of divisions. In practice, we randomly select several such splits, and choose the one which best minimizes Eq.7. As Wu and Nevatia [20] already point out that unbalanced split will impair the performance, we set a threshold to assure that the smaller subset has no less than 30% of the total samples.

### 3.3.2 Division on Multiple Features

When there are multiple features, we first get the best divisions according to each single feature. One division can be viewed as a vector with $M$ elements of 0 or 1, where $M$ is the number of positive samples, and the values indicate to which subset a sample belongs. When there are $T$ features ($T > 1$), we can get $T$ such vectors which make up an $M \times T$ voting table $V$, as shown in Figure 2. Now, we need to make a compromise to get a final partition so that it is most consistent with all divisions on one feature.

An intuitive way of compromise is to count votes for 0 and 1 respectively, and choose the larger one. Yet, as discussed at the beginning of this section, our target is to minimize $\prod_{i=1}^{T} Z_i$, which indicates divisions on single feature do not share equal importance. Once a division $D$ is decided, all $Z_i$ for the two subsets can be computed. Therefore, $Z_i$ is a function of $D$. After the division, we can get two above products, and our target is to minimize the larger one. We can rewrite the target as follows,

$$\text{argmin}_D \max\left\{ \prod_{i=1}^{T} Z_{i0}(D), \prod_{i=1}^{T} Z_{i1}(D) \right\}$$

$$= \max\left\{ \sum_{i=1}^{T} (-\ln Z_{i0}(D)), \sum_{i=1}^{T} (-\ln Z_{i1}(D)) \right\}$$

(10)

where $Z_{i0}(D)$ and $Z_{i1}(D)$ denote $Z_i$ in the two subsets.

Equ.10 indicates that during the compromise, partitions on different features should have different weights, and $-\ln Z_i$ is just a proper definition. Thus, deciding which subset one sample belongs to becoming a weighted voting problem, and each partition can be viewed as a weighted vote for subset 0 or 1. We sum up all the votes for the two clusters from all partitions, and choose the larger one as the final vote for one sample $x_k$, i.e.,

$$\text{Vote}_k = \max\left\{ \sum_{i: V_{ki}=0} -\ln Z_{i0}, \sum_{j: V_{kj}=1} -\ln Z_{j1} \right\}$$

(11)

where $V_{ki}$ is the element in the voting table $V$ at row $k$ and column $i$.

We should note that the larger the vote is, more partitions agree on the final label. Therefore, to best satisfy most partitions on single feature, we aim at maximizing the sum of votes for all positive samples as

$$\text{Vote}_{all} = \sum_{k=1}^{M} \text{Vote}_k$$

(12)

For one partition, if we switch the subset label 0 with 1, the $Z$ value does not change. However, assignment of subset id may have great influence on the overall performance. Let us suppose that there are two features and four samples, and the partition vectors on feature 1 and feature 2 are 0110 and 1001 respectively. Then, it seems that the two partitions have no common agreement at all. However, if we switch subset labels for the second partition, it will not change $Z$ value of partition 2, but will have a full agreement with partition 1. Therefore, we try to switch labels of subsets one by one on all partitions. As shown in Figure 2, for the red column, we compare the Vote$_{all}$ values before and after switch 0 with 1. If the switch enlarge Vote$_{all}$, we make the switch; otherwise, keep the original labels. By such process, we heuristically maximize Vote$_{all}$, and satisfy most partitions on single feature.

There are several differences between our approach and k-means in CBT. First, we aim at benefiting boosting results by minimizing training loss, while k-means only clusters samples by feature similarity, which does not necessarily
improve boosting results. Secondly, our method not only considers distributions of positive samples, but also takes distributions of negative ones into account by Equ.7, which further improves the discriminability of classifiers. Furthermore, our approach chooses different weights for different features according to their powers, which better makes use of characteristics of features than same weights method used in k-means does.

For efficiency purpose, our heuristic algorithm for minimizing the overall training loss is achieved by minimizing that on each feature. Although such approach is an approximation, experiments show that our method minimizes the original target function well.

4. Experiments

As human detection is an important problem, many datasets are available, such as MIT [12], INRIA [4], CAVIAR [3], and Caltech [5]. However, all of these datasets mainly focus on pedestrians in standing or walking poses, lacking humans in other highly articulated poses. Therefore, we build our own dataset from 26 infrared surveillance videos, which are captured by a Thermal-Eye 250D camera with a spectral response in 7-14 microns. In these videos, people are acting naturally with heavy articulations. We cut more than 2000 positive samples and select over 1000 background images from 9 long video clips. For test data, we cut more than 2500 frames from other 17 video clips, including 4627 humans in them.

In our dataset, the groundtruths of humans are not a fixed normalized size, i.e. fixed width/height ratio, which is common in some other pedestrian datasets. The reason is that bounding boxes of articulated humans may change much more with poses than those of pedestrians, and a fixed width/height ratio is not precise enough. This produces a problem for cropping training samples, which should be normalized to the same size. If all samples are cropped with a uniform reference window size, many of them will inevitably include much background area. Learning on such noisy data may lower the training efficiency and suffer from overfitting. Therefore, in our experiment, we first do a k-means clustering on the width/height ratios of all positive samples, and put them into $p$ clusters, so that samples in the same cluster have similar ratios. Then we cut samples with the mean width/height ratio of corresponding clusters, and train one VCBT for each. Thus, background will occupy only a small area of the samples. Figure 4 shows some training data with $p=2$ and $p=4$. We will later show the effects of different cluster number $p$.

In our experiments, we adopt the Edgelet feature [21], which already shows good performance on pedestrian detection. As Edgelet features come from silhouette patterns, our method is not restricted to infrared data, but is also generalizable to color data.

In this section, we will first compare the training and testing errors of our VCBT with those of CBT. Then we will compare our detection results with CBT and Zhang et al’s cascade SVM [23].

4.1. Comparisons of Split Strategies

First, we compare our sample split strategy with that used in CBT [20], i.e., k-means. Since our method aims at minimizing training loss, i.e., $\prod_{t=1}^{T} Z_t$, we fist compare this value. As our method adopts random sampling techniques, we need to repeat one experiment for several times. In each iteration, we randomly chose 70% positive samples and 70% negative samples as training data, and trained our VCBT and CBT both on them. In our experiment, we set the same splitting threshold for VCBT and CBT. The sample split occurs only when there are three consecutive weak classifiers whose $Z$ values are larger than 0.97. Therefore, the first splitting point will be the same in our VCBT and CBT. However, owing to different sample division methods, later splitting points may be different. A fair comparison should be splitting on the same data. Thus, we stop the training process when the first split and retraining finish, and compare the upperbound of training errors before and after the split. As there will be two $\prod_{t=1}^{T} Z_t$ after the split, we only choose the larger one, which is consistent with our minimizing function in Equ. 7. In this experiment, we set the pre-cluster number $p$ as two, and compare results on the two subsets respectively. The results are shown in Figure 5.

From Figure 5, we can see that our method lowers the training error upper bound after the split; the red curves are always below the blue ones. This indicates that our split
method successfully reduces the training loss, and therefore facilitates the boosting process to learn more powerful classifiers for each subset. In addition, our split method also outperforms simple k-means used in CBT. This is because our split strategy heuristically optimizes the sample division in boosting process by considering distributions of both positive and negative samples, whereas k-means only cluster samples on feature similarity without considering the boosting process or different importance of these features.

As low training loss is not necessarily equal to low testing errors, we designed a cross validation experiment. Similar to the above experiment, we randomly select 70% positive samples and 70% negative samples as training data, and use others as test data. After the first split and retraining, we set a fixed threshold for negative sample rejection rate, and compare the pass rate of positive samples. The results are shown in Figure 6.

Figure 6 shows that our split method performs better than k-means. For different negative reject rates, positive pass rates in our method are generally higher than those for CBT. Although our split method aims at minimizing training loss, it also performs well on test data.
4.2. Results of VCBT and CBT

Next, we compare our final detection results with those of CBT. To test the effects of different pre-cluster number \( p \), we produce three precision-recall curves by setting \( p \) to 1, 2, and 4, which are shown in Figure 7.

From Figure 7, we find the best performance is reached when \( p = 2 \), while decreasing \( p \) to 1 or increasing \( p \) to 4 both impair the performance. When \( p \) is set to 1, no pre-cluster is made. All samples are cut to the same normalized size. However, as discussed in previous sections, articulated humans may have quite different width/height ratios of bounding boxes. Using the same size will inevitably introduce much background for some samples, and background region may be even larger than human region. On the other hand, when \( p \) is set large, each cluster will has one VCBT or CBT, and in addition there may be several splits in each VCBT or CBT. Thus, the total number of classifiers will be large, and all their false alarms will sum together. Furthermore, as discussed in previous sections, articulated humans may have quite different width/height ratios of bounding boxes. Using the same size will inevitably introduce much background for some samples, and background region may be even larger than human region. On the other hand, when \( p \) is set large, each cluster will has one VCBT or CBT, and in addition there may be several splits in each VCBT or CBT. Thus, the total number of classifiers will be large, and all their false alarms will sum together. Furthermore, we can see that for all three values of \( p \), our VCBT always outperforms CBT. As shown in the last experiment, our method has lower training and testing error on each split, and therefore continuously performs better than CBT. The predominance finally shows in the detection performance.

4.3. Results of VCBT and Cascade SVM

Finally, we compare our VCBT with the Cascade SVM in [23]. Zhang et al [23] first cluster samples according to their silhouettes, so that human samples in each cluster have similar poses. Then a cascade SVM is trained on each cluster. To deal with background area in training samples, they use average silhouettes in each cluster to emphasize features on human parts. We test different numbers of pre-clusters, and the results are shown in Figure 8.

From Figure 8, we find that our implementation of the Cascade SVM performs not so well, and has a much lower precision-recall curve than VCBT method does. There are several reasons for this. In Zhang et al’s method, each cluster has one cascade SVM, while we find some SVMs produce many false alarms on test data. Although most SVMs may perform well, a few bad ones will damage the final results a lot. In addition, as many clusters are necessary to ensure the similarity of samples in the same subset, the number of positive samples in one cluster is often insufficient, which may result in overfitting.

Finally, Figure 9 shows some examples of our detection results. We can see that our method is able to find most highly articulated persons and can deal with moderate occlusions, but often has miss detections for heavy occlusions. Miss detections may also occur when humans are small.

5. Conclusion

In this paper, we propose an effective sample division approach and incorporate it into the cluster boosted tree to detect humans with highly articulated poses. At each split point in the tree, a heuristic algorithm is adopted to obtain better sample divisions in terms of classification errors on the divided samples. This algorithm first finds optimal sample divisions independently according to each feature, and then makes a compromise between them to achieve a final division by a voting like process that considers different importance of features. Experiments show that our approach of dividing samples improves the detection accuracy compared to the conventional k-means clustering method. This approach does not restrict on specific features or even specific problems, and thus can be generalized to other complicated object detection problems.

6. Acknowledgement

This research was supported, in part, by the Office of Naval Research under Contract #N00014-06-1-0470.
Figure 9. Examples of highly articulated human detection results. Correct detections, false alarms, and miss detections are labeled with green, red, and yellow respectively.

References


