MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan
10-4

Pedestrian Detection by Boosting Soft-Margin SVM with Local Feature Selection

Kenji NISHIDA
Neuroscience Research Institute
National Institute of Advanced
Industrial Science and Technology
(AIST)
Central 2, 1-1-1 Umezono, Tsukuba,
IBARAKI 305-8568 JAPAN
kenji.nishida@aist.go.jp

Takio Kurita
Neuroscience Research Institute
National Institute of Advanced
Industrial Science and Technology
(AIST)
Central 2, 1-1-1 Umezono, Tsukuba,
IBARAKI 305-8568 JAPAN
takio-kurita@aist.go.jp

Abstract

We present an example-based algorithm for detecting objects in images by integrating component-based classifiers, which automaticaly select the best local-feature for each classifier and are combined according to AdaBoost *algorithm. The system employs soft-margin SVM for base learner, which is trained for all local-features and the optimal feature is selected at each stage of boosting. The proposed method is applied to the MIT CBCL pedestrian image database, and shows fairly good result with 10 local-features such as full image, upper half, lower half, right half, left half, head, right arm, left arm, legs, and body center.*

1 Introduction

In this paper, we present an example-based algorithm for detecting objects in images by integrating component-based classifiers, which automatically select the best local-feature for each classifier and are combined according to *AdaBoost*[1] algorithm. Our method can be applied to any object composed of distinct identifiable parts that are arranged in a well-defined configuration, such as cars and faces. However we focus on the pedestrian detection in images, which could be used in driver assistance systems and video surveillance systems. Pedestrian (people) detection is more challenging than detecting other objects such as cars and faces, since people take a variety of shapes and it is nontrivial to define a single model that captures all of these possibilities.

Gavrila[3] employed hierarchical template matching to find pedestrian candidates from incoming images. The method previously provide multiple templates which are outline egde image of typical pedestrians and dissimilarity (or similarity) between the edge feature of incoming images are measured by chamfer distance. In this method, the shape variety of the pedestrians are accommodated with variety of templates, which would bound the system performance.

Mohan et. al[2] applied an Adaptive Combination of classifiers (ACC) to pedestrian detection. Their system consists of two stage hierarchical classifiers. The first stage is structured with four distinct example-based classifiers, which separately trained to detect different component of pedestrians, such as the head, legs, right arm, and left arm. The second stage has an example based classifier which combines the result of the component detectors in first stage to classify a pattern as either a "person" or a "non-person". Support Vector Machine (SVM)[5, 6, 7] is employed for each classifier. Their result indicates that combination of component based detectors perform better than a full-body person detector. In their system, the components are determined in advance and they are not exactly optimal for classification of the examples.

Viola et. al[4] presented a pedestrian detection system which integrates image intensity information with motion information. Their detection algorithm scans a detector over two consecutive frames of video sequence, and the detector trained using *AdaBoost* to take advantage of both motion and appearance information. They achieved high detection speed (about four frames/second) and very low false positive rate, with combining two different modality of information to one detector.

Though Viola showed the advantage of integrating motion information, it is still difficult to apply their algorithm to on-board pedestrian detection system, since canceling a movement of the camera is difficult only from visual information. Therefore, we focused on pedestrian detection from static images to realize our example-based object detection method. We employ soft-margin SVM for base learners of *AdaBoost*, and the best local-feature is automatically selected at each stage of boosting. Our preliminary result shows that proposed method achieves fairly good classification ratio.

We describe our object detection method in next section, and the preliminary result is introduced in following section. In final section, we give some theoretical consideration on our result.

2 System Configuration

2.1 Algorithm

Fig. 1 shows the overall algorithm of our pedestrian detection system, which is based on AdaBoost and Local-Feature selection. We define a local-feature as a portion of an input vector, usualy extracted with some meanings such as sub-region of an image (fig. 2). The contribution of local-feature selection is performed as selecting the best local-feature at each boosting step. At ith boosting step, the ith base learner is trained under the sample weights determined by $i-1$th base learner for all the local features in Local-Feature pool, and selects the best local-feature (which has the lowest error ratio) for ith base learner. The sample weights for next boosting step and significance of the base learner is computed according to the error ratio.

We employ MIT CBCL database for sample data, which contains 926 pedestrian images with 128×64 pixel, and we collected 2,000 random non-pedestrian images. We reduced the resolution of all the samples to 64×32 before we applied them to our system. We used the intensity of the pixels as an input vector, and the local-features are extracted as adequate sub-regions of the input images. We took ten local features for our evaluation; such as whole pedestrian body, upper half, lower half, left half, right half, head, left arm, right arm, legs, and center of body. However, the number and the size of local-features can arbitrary be determined. Fig. 2 shows the original images and local features. We selected 700 pedestrian images and 700 non-pedestrian images for training, and 200 pedestrian images and 200 non-pedestrian images for testing the generalization error.

2.2 Boosting Soft-Margin SVM

We employed a soft-margin SVM for a base learner. We first describe a SVM briefly and a description on soft-margin SVM is followed.

When the classification function is given as

$$y = \mathrm{sign}(\boldsymbol{w}^T\boldsymbol{x} - h) \tag{1}$$

where $\boldsymbol{x}$ stands for an input vector, $\boldsymbol{w}$ stands for a weight vector of the input, and h stands for a threshold. Function sign(u) is a sign function, which ouputs 1 when $u > 0$ and outputs -1 when $u \leq 0$. A SVM determines a separating hyperplane with maximal margin (distance), which is the distance between the separating hyperplane and a nearest sample. If the hyperplane is determined, there exists a parameter to satisfy

$$t_i(\boldsymbol{w}^T\boldsymbol{x_i}) \geq 1, \;\; i = 1, \ldots, N. \tag{2}$$

This means that the samples are separated by two hyperplane H1: $\boldsymbol{w}^T\boldsymbol{x}_i - h = 1$ and H2: $\boldsymbol{w}^T\boldsymbol{x}_i - h = -1$, and no samples exist between them. The distance between separating hyperplane and H1 (or H2) is defined as $1/\|\boldsymbol{w}\|$. Determining the parameters $\boldsymbol{w}$ and

---

1. Let N be the number of samples, M be the number of boosting steps, L be the number of local-features in Local-Feature pool.
2. Generate Local-Feature pool from input samples $\boldsymbol{x}$, such as $\boldsymbol{x} \rightarrow \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^L$.
3. Initialize the observation weights $w_i = 1/N$, $i = 1, 2, \ldots, N$.
4. For $m = 1$, to M:
   (a) For $l = 1$ to L
      i. Fit a classifier $G_m^l(x^l)$ to the training samples of local-feature x^l that are randomly selected depending on the weights w_i from all the training samples
      ii. Compute $err_m^l = \frac{\sum_{i-1}^{N} w_i I(y_i \neq G_m(x_i^l))}{\sum_{i=1}^{N} W_i}$.
   (b) set err_m with the smallest err_m^l, $l = 1, 2, \ldots, L$.
   (c) set $G_m(x) \leftarrow G_m^l(x^l)$ with l in above step.
   (d) compute $\alpha_m = \log((1 - err_m)/err_m)$.
   (e) set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$, $i = 1, 2, \ldots, N$.
5. Output $G(x) = \mathrm{sign}[\sum_{m=1}^{M} \alpha_m G_m(x)]$.

Figure 1: AdaBoost with Local-Feature selection

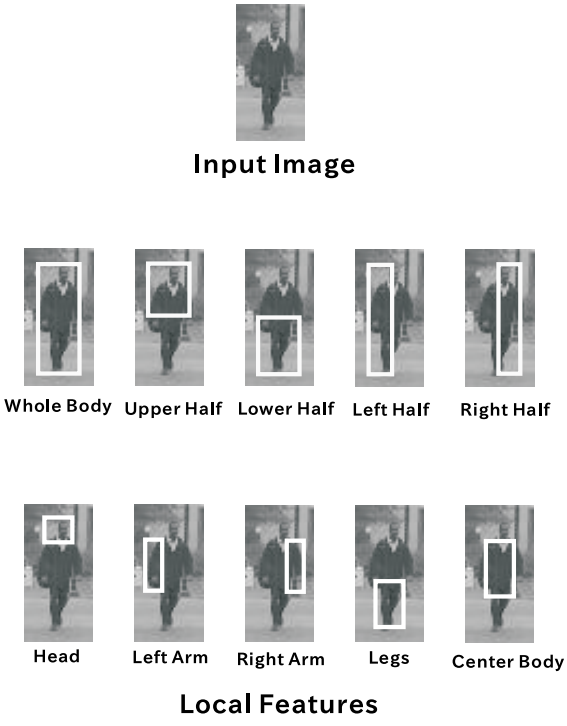

Figure 2: Sample Images and Local Features

h which give maximal margin is defined as an optimization problem of the following evaluation function

$$L(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{3}$$

under a constraint

$$t_i(\boldsymbol{w}^T\boldsymbol{x}_i - h) \geq 1, \;\; i = 1, \ldots, N \tag{4}$$

where t_i stands for the correct class label for an input vector x_i.

A soft-margin SVM allows some training samples to violate the hyperplanes H1 and H2. When a distance from the H1 (or H2) is defined as $\xi_i/\|\boldsymbol{w}\|$ for the violating samples, the sum

$$\sum_{i=1}^{N} \frac{\xi_i}{\|\boldsymbol{w}\|} \tag{5}$$

should be minimized. Therefore, a soft-margin SVM is defined as an optimization problem of the following evaluation function

$$L(\boldsymbol{w}, \boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \xi_i \tag{6}$$

under a constraint

$$\xi_i \geq 0, \;\; t_i(\boldsymbol{w}^T\boldsymbol{x}_i - h) \geq 1 - \xi_i, \;\; i = 1, \ldots, N \tag{7}$$

where C stands for a **cost** parameter for violating hyperplane H1 (or H2). Solving this problem with an optimal solution $\boldsymbol{\alpha}^*$, the classification function can be redefined as

$$\begin{aligned} y &= \mathrm{sign}(\boldsymbol{w}^{*T}\boldsymbol{x} - h^*) \\ &= \mathrm{sign}(\sum_{i \in S} \alpha_i^* t_i \boldsymbol{x}_i^T \boldsymbol{x} - h^*). \end{aligned} \tag{8}$$

The samples are grouped with α_i^*; a sample x_i is classified correctly when $\alpha_i^* = 0$, when $0 < \alpha_i^* < C$ the sample x_i is also classfied correctly and it locates on the hyperplane H1 (or H2) as a support-vector, if $\alpha_i^* = C$ the sample x_i becomes a support-vector but it locates between H1 and H2 with $\xi \neq 0$.

The Kernel-Trick, which drastically improved the performance of SVM, can also be applied to soft-margin SVM. In Kernel-Trick, the input vectors are transformed by non-linear projection $\phi(\boldsymbol{x})$ and linearly classified in the projected space. Since SVM depends on the product of two input vectors, the product of the input vectors in projected space can be used instead of computing the non-linear projection of the each input vector, such as

$$\phi(\boldsymbol{x}_1)^T\phi(\boldsymbol{x}_2) = K(\boldsymbol{x}_1, \boldsymbol{x}_2). \tag{9}$$

K is called a *Kernel Function* and usualy selected a simple function, Gaussian function

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left(\frac{-||\boldsymbol{x}_1 - \boldsymbol{x}_2||^2}{2\sigma^2}\right) \tag{10}$$

Table 1: The Error Ratio Comparison

	Error Ratio
Gavrila	10-40%*
Mohan	1-2%
Viola	10%
Our Result	2-3%

* For first stage.

for instance. The classification function can be redefined by replacing input vectors with kernel functions, as follows

$$\begin{aligned} y &= \mathrm{sign}(\boldsymbol{w}^{*T}\phi(\boldsymbol{x}) - h^*) \\ &= \mathrm{sign}(\sum_{i \in S} \alpha_i^* t_i \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}) - h^*) \\ &= \mathrm{sign}(\sum_{i \in S} \alpha_i^* t_i K(\boldsymbol{x}_i, \boldsymbol{x}) - h^*). \end{aligned} \tag{11}$$

Introducing cost parameter C, we could have two choice to realize the sample weighing in AdaBoost, one is building SVM with defining a cost parameter as a weight of each sample, the other is re-sampling according to the sample weights. Schwenk et. al[8] showed that defining a pseudo-loss function and re-sampling have similar effect in AdaBoost, we therefore took re-sampling so that we could use LIBSVM[9] for our evaluation. 1,000 images are re-sampled from 1,400 images in training samples.

3 Results

As described in previous section, the base learners are trained for 1,000 images re-sampled from 1,400 training images, and tested by 400 images, while varying the soft-margin cost parameter.

Fig. 3 shows the error ratio against the number of boosting steps with cost parameter C for 0.1, 0.7, and 100. All ten local features are almost evenly selected at each boosting steps. The test error is higher than 11% with C=0.1 and 7% with C=100 without boosting. It continuously decreases according to the boosting steps, even after the training error converges to zero for C=0.7 and C=100. After 100 boosting steps, test error reaches to 4% with C=0.7, 4.5% with C=100, and 5.25% with C=0.1. This result indicates that the boosting improves the generalization error and does not behave as over-training.

Table 1 shows the comparison of error ratio against the former researches. Our result achieved better error ratio against Gavrila and Viola, while little bit worse than Mohan. Considering the difference of non-pedestrian data, we would conclude that our result showed the almost same performance as Mohan, Papageorgiou and Poggio.

Fig. 4 shows the test error against cost parameter C after 50 boosting steps. The test error records the minimal value of 4% at c=0.7. This result indicates that

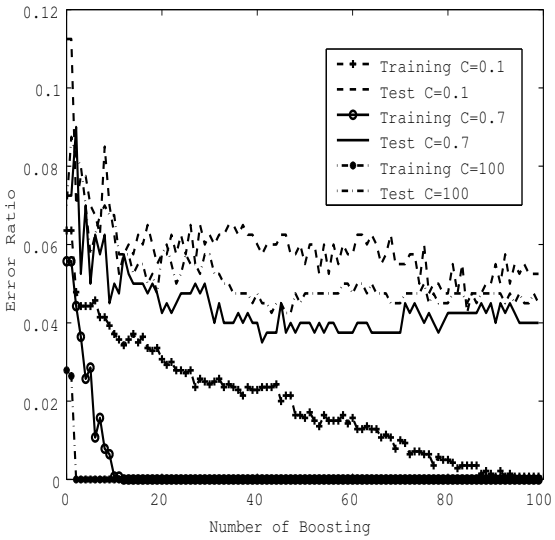

Figure 3: Error Ratio against Number of Boosting

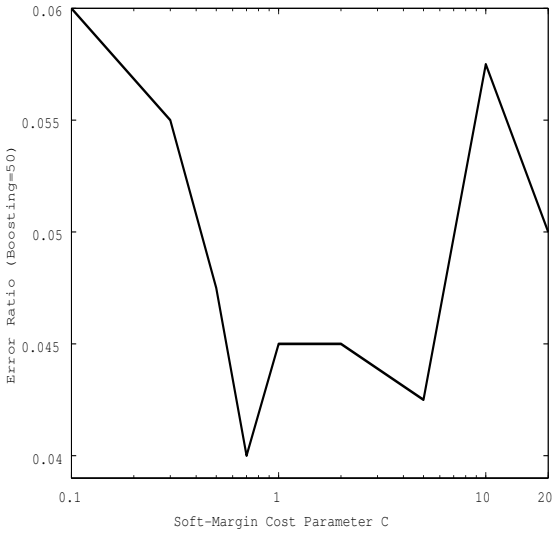

Figure 4: Error Ratio against Cost Parameter C

soft-margin SVM advantageous to usual hard-margin (or firm-margin) SVM for boosting, however there exists the optimal value for cost parameter C.

4 Discussion

Schwenk empirically showed that boosting does not affect the over-training in [8], and Schapire showed the margin of each base learner bounds the final boosted test error in [10]. Our result follows their insight, and can be reconsidered with Schapire's idea of the margin of base learners.

SVM maximizes the distance between H1 and H2, as described in section 2. Introducing soft-margin, the distance can be increased, that means the first term of equation 6 can be decreased. However, if soft-margin allows too many samples to violate H1 (or H2), the second term of equation 6 increases. Therefore, the cost parameter for soft-margin should be set some optimal value to minimize the whole equation 6.

5 Conclusion and Future Work

We presented the object detection method by boosting soft-margin SVM with local-feature selection. The experimental result showed fairly good generalization error ratio of 4%. In this paper, we focused on pedestrian detection using only pixel intensity as a feature, although the other features, such as edge, sobel filtering or chamfer distance, are thought to improve the classification performance. These different kind of features can easily be integrated in our local-feature selection phase, thus we are planning to examine the integration of multiple features in our system.

The result also gave us an insight on the optimal soft-margin cost parameter in SVM classifier. We will have some more consideration on this subject.

References

[1] T.Hastie, R.Tibshirani, J.Friedman, *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*, Springer-Verlag, 2001.

[2] A.Mohan, C.P.Papageorgiou and T.Poggio, "Example-Based Object Detection in Images by Components," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vo.23, No.4, pp.349-361, 2001.

[3] D.M.Gavlira, "Pedestrian Detection from a Moving Vehicle", *Proc. of Europian Conference on Computer Vision*, pp.37-49, 2000.

[4] P.Viola, M.J.Jones and D.Snow,"Detecting Pedestrians Using Patterns of Motion and Appearance", *Proc. of Int'l Conf. on Computer Vision*, pp.734-741, 2003.

[5] V.N.Vapnik, *Statistical Learning Theory*, John Wiley & Sons (1998).

[6] B.Scholkopf, C.J.C.Burges, A.J.Smola, *Advances in Kernel Methods - Support Vector Learning*, The MIT Press, 1999.

[7] N.Cristianini, J.S-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[8] H.Schwenk and Y.Benjio, "Boosting Neural Networks", *Neural Computation*, Vol.12, pp.1869-1887, 2000.

[9] C.C.Chung and C.J.Lin, "LIBSVM: a libraly for support vector machines", *Software available at* `http://www.csie.ntu.edu.tw/`, 2001.

[10] R.E.Schapire, "The Boosting Approach to Machine Learning An OverView", *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.