Labeling Image Patches by Boosting based Median Classifier

Songfeng Zheng
SongfengZheng@MissouriState.edu

Department of Mathematics
Missouri State University
901 S. National Ave.
Springfield, MO 65897, USA

Abstract

This paper presents a median based classifier which predicts the conditional median of the class label given the feature vector. The class label is defined through a hidden variable whose median is further modeled as an additive model of the feature vector. We propose to estimate the conditional median of the hidden variable given the feature vector in the framework of generic functional gradient algorithm [5]. An equivalent form of the definition of median is introduced, whose smoothed version is employed as the objective function. To fit the model for the conditional median, the proposed objective function is maximized by gradient ascent in functional space, updating the fitted model a small step in the gradient direction in each iteration. The resulting algorithm, Median Boost, is a boosting like procedure which obtains the informative features and the classifier at the same time. On the task of labeling building blocks in natural images, the comparison results show that Median Boost performs better than or similar to several alternatives.

1 Introduction

Various classification methods have been widely applied to computer vision problems, for example, AdaBoost [16], Support Vector Machines [12, 14], logistic regression [13]. If the feature vector has long tail distribution, or there are outliers in the data, it is well-known that the median is usually more stable than other statistics [7, 8, 10, 11]. In particular, if some of the components of the feature vector have high variability then, regardless of whether or not these components convey information about the class label, classification accuracy might be poor. In computer vision problems, the aforementioned case is possible to happen since the data are often in very high dimensional spaces, and there is ambiguity in labeling the training data which could introduce outliers. As such, it is worthwhile trying to design classifiers based on the median information. To the best of our knowledge, median-based classifiers have not been reported for classification problems in computer vision.

This paper proposes to estimate the conditional median of class label given the feature vector in the framework of the generic functional gradient algorithm [5]. Toward this purpose, we define the class label through a hidden variable whose median is further modeled as an additive model of the feature vector. An equivalent form of the definition of median is introduced, whose smoothed version is employed as the objective function. In functional space, gradient ascent is performed to maximize the proposed objective function, updating the fitted model (i.e., the estimated median of hidden variable) a small step in the gradient direction in each iteration. Finally, the class label is predicted using the estimated conditional median of the hidden variable. The resulting algorithm is a boosting like procedure, which is called Median Boost. Similar to AdaBoost [4], Median Boost obtains the informative features and the classifier simultaneously.

The proposed Median Boost was tested on a publicly available dataset for labeling building blocks in natural images [12, 13], and we compared the obtained results to those obtained by the median based classifier reported in [7] and the Probabilistic AdaBoost Cascade [18], which is a variation of Adaboost Cascade [16]. The results show that compared to the alternative methods, Median Boost achieves better or similar performance in terms of detection rate and site-wise error rate.

2 Boosting as Functional Gradient Descent

Boosting [4] is a classic algorithm in pattern classification and is well known for its simplicity and good performance. The powerful feature selection mechanism of boosting makes it suitable to work in very high dimensional spaces. Moreover, boosting can pick informative features and obtain the final classifier simultaneously. Friedman et al. [5, 6] developed a general statistical framework which yields a direct interpretation of boosting as a method for function estimation, which is a "stagewise, additive model".

Consider the problem of function estimation

$$f^*(\mathbf{x}) = \arg\min_f \mathrm{E}[l(Y, f(\mathbf{x}))|\mathbf{x}],$$

where $l(\cdot,\cdot)$ is a loss function which is typically differentiable and convex with respect to the second argument. Estimating $f^*(\cdot)$ from the given data $\{(\mathbf{x}_i, Y_i),\ i = 1, \cdots, n\}$ can be performed by minimizing the empirical risk $n^{-1}\sum_{i=1}^n l(Y_i, f(\mathbf{x}_i))$ and pursuing iterative steepest descent in functional space. This leads us to the generic functional gradient descent algorithm [1, 5], as shown in Algorithm 1.

Algorithm 1 Generic Functional Gradient Descent Algorithm

```
0: Set $m = 0$ and initialize $f^{[0]}(\cdot) = 0$.
1: Increase $m$ by 1, compute the negative gradient $-\frac{\partial}{\partial f} l(Y, f)$ at $f^{[m-1]}(\mathbf{x}_i)$:
       $$U_i = -\frac{\partial l(Y_i, f)}{\partial f}\bigg|_{f=f^{[m-1]}(\mathbf{x}_i)}, \quad i = 1, \cdots, n.$$
2: Fit the negative gradients $U_1, \cdots, U_n$ to $\mathbf{x}_1, \cdots, \mathbf{x}_n$ by the base procedure:
       $$\{(\mathbf{x}_i, U_i),\ i = 1, \cdots, n\} \longrightarrow g^{[m]}(\cdot).$$
3: Update the estimation by $f^{[m]}(\cdot) = f^{[m-1]}(\cdot) + \nu g^{[m]}(\cdot)$, where $\nu$ is a step-length factor.
4: Check the stopping criterion, if not satisfied, go to step 1.
```

Many boosting algorithms can be understood as functional gradient descent with appropriate loss function. For example, if we choose $l(Y, f) = \exp(-(2Y-1)f)$, we would recover the AdaBoost algorithm [6], and L_2-Boost [2] corresponds to $l(Y, f) = (Y - f)^2/2$.

3 Median Classifier and Median Boost

This section derives the proposed Median Boost algorithm, and discusses the connection to the existing methods.

3.1 Predicting Median of Binary Variable

Consider the following model:

$$Y^* = h(\mathbf{x}) + \varepsilon \quad \text{and} \quad Y = I(Y^* \geq 0),$$

where Y^* is a continuous latent variable, $h(\cdot)$ is the true model for Y^*, ε is a disturb, and $Y \in \{0, 1\}$ is the observed label given the feature vector $\mathbf{x} \in \mathbf{R}^p$, $I(\cdot)$ is the indicator function with $I(\cdot) = 1$ if the condition is true, otherwise $I(\cdot) = 0$.

Let $M(Y^*|\mathbf{x})$ be the conditional median of Y^* given $\mathbf{x}$, and let $g(\cdot)$ be a real monotone increasing function. Clearly, for any value of y

$$P(Y^* \geq y|\mathbf{x}) = P\left(g(Y^*) \geq g(y)|\mathbf{x}\right),$$

it follows that

$$g\left(M(Y^*|\mathbf{x})\right) = M(g(Y^*)|\mathbf{x}). \tag{1}$$

Since indicator function $I(t \geq 0)$ is monotone increasing with respect to t [9, 11], Eqn. (1) indicates that

$$I\left(M(Y^*|\mathbf{x}) \geq 0\right) = M(I(Y^* \geq 0)|\mathbf{x}) = M(Y|\mathbf{x}). \tag{2}$$

Therefore, the conditional median of Y, $M(Y|\mathbf{x})$, could be obtained by fitting the conditional median of Y^*, $M(Y^*|\mathbf{x})$. If we model the median of the latent variable Y^* by a function $f(\mathbf{x}, \beta)$ with β as the parameter vector, i.e.,

$$M(Y^*|\mathbf{x}) = f(\mathbf{x}, \beta),$$

it follows from Eqn. (2) that the conditional median of the binary variable Y can be written as

$$M(Y|\mathbf{x}) = I\left(f(\mathbf{x}, \beta) \geq 0\right). \tag{3}$$

Recall that, if Z is a random variable, then the median of Z can be defined through the following minimization problem (refer to [8] for a proof):

$$\text{median}(Z) = \arg\min_c \mathrm{E}(|Z - c|).$$

As such, given the training data $\{(\mathbf{x}_i, Y_i), i = 1, \cdots, n\}$, with $\mathbf{x}_i \in \mathbf{R}^p$ and $Y_i \in \{0, 1\}$, to fit the model for the conditional median of the class label Y, i.e., to estimate the parameter vector

β in Eqn. (3), we can solve the following minimization problem and let $\mathbf{b}$ be the estimated parameter vector:

$$\mathbf{b} = \arg\min_{\beta} \left\{ L(\beta) = \sum_{i=1}^{n} |Y_i - I(f(\mathbf{x}_i, \beta) \geq 0)| \right\}. \tag{4}$$

It can be verified that Eqn. (4) is equivalent to the following maximization problem:

$$\mathbf{b} = \arg\max_{\beta} \left\{ S(\beta) = \sum_{i=1}^{n} [Y_i - 0.5] I(f(\mathbf{x}_i, \beta) \geq 0) \right\}. \tag{5}$$

3.2 Median Classifier

From the definition of the binary variable $Y = I(Y^* \geq 0)$, it follows that

$$P(Y=1|\mathbf{x}) = P(I(Y^* \geq 0)|\mathbf{x}) = P(Y^* \geq 0|\mathbf{x}). \tag{6}$$

Since $f(\mathbf{x}, \beta)$ is the conditional median of the latent variable Y^*, we have

$$P(Y^* \geq f(\mathbf{x}, \beta)|\mathbf{x}) = 0.5. \tag{7}$$

Thus, if $f(\mathbf{x}, \beta) = 0$, combining Eqns. (6) and (7) yields

$$P(Y=1|\mathbf{x}) = P(Y^* \geq 0|\mathbf{x}) = P(Y^* \geq f(\mathbf{x}, \beta)|\mathbf{x}) = 0.5;$$

if $f(\mathbf{x}, \beta) > 0$,

$$P(Y=1|\mathbf{x}) = P(Y^* \geq 0|\mathbf{x}) > P(Y^* \geq f(\mathbf{x}, \beta)|\mathbf{x}) = 0.5.$$

In summary, we have the following:

$$P(Y=1|\mathbf{x}) \gtreqless 0.5 \quad \text{for} \quad f(\mathbf{x}, \beta) \gtreqless 0, \tag{8}$$

which is an inequality of the posterior probability of the label given the predictor vector. Consequently, once the model is fitted, i.e., the parameter vector $\mathbf{b}$ is obtained, we can make prediction by

$$\hat{Y} = I(f(\mathbf{x}, \mathbf{b}) \geq 0), \tag{9}$$

where $\hat{Y}$ is the predicted label for the input feature vector $\mathbf{x}$. We call Eqn. (9) as a median classifier because it makes decision based on the conditional median of the latent variable. Eqn. (8) inidicates that the median classifier is equivalent to the Bayesian classifier with cut-off posterior probability 0.5.

Median classifier is learned by performing the maximization problem defined in Eqn. (5), which has a close connection to the training error rate. Let each term in Eqn. (5) be

$$S_i = [Y_i - 0.5] I(f(\mathbf{x}_i, \beta) \geq 0) = 0.5 I(Y_i = 1) I(f(\mathbf{x}_i, \beta) \geq 0) - 0.5 I(Y_i = 0) I(f(\mathbf{x}_i, \beta) \geq 0).$$

Then, Eqn. (5) reads

$$\begin{aligned} S(\beta) &= \sum_{i=1}^{n} S_i = 0.5 \sum_{i=1}^{n} I(Y_i = 1) I(f(\mathbf{x}_i, \beta) \geq 0) - 0.5 \sum_{i=1}^{n} I(Y_i = 0) I(f(\mathbf{x}_i, \beta) \geq 0) \\ &= 0.5TP - 0.5FP = 0.5TP - 0.5(Neg - TN) = 0.5TP + 0.5TN - 0.5Neg \end{aligned} \tag{10}$$

where TP, FP, and TN are the true positive number, false positive number, true negative number of the classifier $I(f(\mathbf{x},\beta)\geq 0)$, and Neg is the number of negative examples. Eqn. (10) indicates that maximizing Eqn. (5) is equivalent to maximizing the sum of TP and TN, i.e., the number of correctly classified examples. Hence, maximizing Eqn. (5) implicitly minimizes the training error, and this argument justifies our choice of the objective function.

3.3 Median Boost: Boosting based Median Classifier

The median classifier defined in this paper is learned by maximizing function $S(\beta)$ defined in Eqn. (5). However, $S(\beta)$ is not differentiable because of the used indicator function. To apply gradient based optimization methods, we replace the indicator function $I(f(\mathbf{x},\beta)\geq 0)$ by its smoothed version and solve

$$\mathbf{b}=\arg\max_{\beta}\left\{S(\beta,h)=\sum_{i=1}^{n}[Y_i-0.5]K\left(\frac{f(\mathbf{x}_i,\beta)}{h}\right)\right\}, \tag{11}$$

where h is a small positive number, and $K(t)$ is smoothed version of the indicator function with the following properties:

$$K(t)>0,\ \forall t\in\mathbf{R},\quad \lim_{t\to\infty}K(t)=1,\quad \lim_{t\to-\infty}K(t)=0.$$

In this paper, we take $K(\cdot)$ as the standard normal cumulative distribution function.

We propose to maximize the objective function in Eqn. (11) by gradient ascent in the framework of functional gradient method [5]. The fitted function is updated in the gradient direction in each iteration. Let $f^{[m]}(\cdot)$ be the fitted function at the m-th iteration, similar to the gradient boosting algorithm (Algorithm 1), we obtain the Median Boost algorithm, which is shown as Algorithm 2, where

$$l(Y,f)=[Y-0.5]K(f/h).$$

Let the base procedure be $h(\mathbf{x},\mathbf{a})$ with $\mathbf{a}$ being the parameter vector. Then the third step in Algorithm 2 can be performed by an ordinary least square regression:

$$\mathbf{a}_m=\arg\min_{\mathbf{a}}\sum_{i=1}^{n}\left[U_i-h(\mathbf{x}_i,\mathbf{a})\right]^2,$$

hence the function $g^{[m]}(\mathbf{x})=h(\mathbf{x},\mathbf{a}_m)$ can be regarded as an approximation of the gradient in the functional space. Step 4 performs gradient ascent in the functional space, in which the step-length factor ν can be determined by line search (e.g., Fibonacci search, Golden section search [17]). Alternatively, for simplicity, in each iteration, we could update the fitted function $f^{[m-1]}(\cdot)$ by a fixed but small step in the gradient direction. To guarantee the performance of the resulting model, ν is fixed at a small value as suggested by [1, 5].

The median classifier has close connection to binary quantile regression [10, 11], which optimizes an objective function similar to Eqn. (5). However, in binary quantile regression, simulated annealing algorithm is employed to perform the optimization. Although a local optimum is guaranteed, simulated annealing is well known for its expensive computation and it is usually difficult to tell when the algorithm converges. As a comparison, Median Boost is based on gradient ascent, which yields a local maximum and converges fast.

Algorithm 2 Median Boost Algorithm

```
0: Given data $\{(\mathbf{x}_i, Y_i),\ i = 1, \cdots, n\}$ with $\mathbf{x}_i \in \mathbf{R}^p$ and $Y_i \in \{0, 1\}$; initialize $f^{[0]}(\mathbf{x}) = 0$.
1: for $m = 1$ to $M$ do
2:     Compute the gradient $\frac{\partial}{\partial f} l(Y_i, f)$ at $f^{[m-1]}(\mathbf{x}_i)$:
           $$U_i = \left.\frac{\partial l(Y_i, f)}{\partial f}\right|_{f = f^{[m-1]}(\mathbf{x}_i)} = \frac{Y_i - 0.5}{h} K'\left(\frac{f^{[m-1]}(\mathbf{x}_i)}{h}\right), \quad i = 1, \cdots, n.$$
3:     Fit the gradients $U_1, \cdots, U_n$ to $\mathbf{x}_1, \cdots, \mathbf{x}_n$ by the base procedure:
           $$\{(\mathbf{x}_i, U_i),\ i = 1, \cdots, n\} \longrightarrow g^{[m]}(\cdot).$$
4:     Update the estimation by $f^{[m]}(\cdot) = f^{[m-1]}(\cdot) + \nu g^{[m]}(\cdot)$, where $\nu$ is a step-length factor.
5: end for
6: Output the classifier $I(f^{[M]}(\mathbf{x}) \geq 0)$.
```

Due to the expensive computation of simulated annealing, binary quantile regression can only work in very low dimensional spaces. However, in applications, we frequently face hundreds, even thousands of features, and it is often desired to find out the informative ones. Clearly, in this case, binary quantile regression is not applicable. On the contrary, Median Boost is designed to work in high dimensional spaces, and by using certain types of base learner (e.g., decision stump [16]), it enables us to select the most informative features.

Hall et al. [7] proposed a median based classifier which works in high dimensional space. For a given feature vector, [7] calculates the L_1 distances from the new feature vector to the component-wise medians of the positive examples and negative examples in the training set, and assigns class label as the class with the smaller L_1 distance to the new feature vector. Although computationally efficient, this simple nearest neighbor like algorithm cannot perform feature selection as the proposed Median Boost.

4 Experiments

We test the proposed Median Boost on the task of labeling building blocks in natural images [12, 13]. The training and testing sets contain 108 and 129 images, respectively, each of size 256×384 pixels. Each image is divided into non-overlapping 16×16 image patches. The ground truth was generated by manually labeling every image patch as *building* or *non-building*. There are 5,203 building patches and 36,269 non-building patches in the training set, and 6,372 building patches and 43,164 non-building patches in the testing set.

4.1 Features

For the building block labeling problem, we use the features described in [12, 13] as our first set of features, which are based on the weighted histogram of the gradient orientation. Please refer to [12] for more details. We also use different combinations (sum, difference, etc.) of features from [12].

We apply different filters (e.g., Gabor filters, Gaussian filters, Canny Edge detectors) to the original image, and extract other features from the filter responses. We notice that most building regions are relatively smooth with small variances while most background regions have cluttered pattern with large variations. This observation inspires us to use mean and variance values of different filter responses (include the original image) inside sub-windows as features.

It is known that histograms of different filter responses can capture texture information [19]. Therefore, we introduce histogram based features, expecting that they can capture the difference between background and the building patches. Inside each sub-window, we calculate the histograms from each filter response, and use each bin of the histogram as a feature; the entropy of the histograms is used as a feature as well to evaluate the regularity of the sub-window.

We further notice that building patches are primarily characterized by straight lines with horizontal or vertical direction, and this motivates us to extract features from the edge map. In canny edge maps with different scales, we count the numbers of horizontal and vertical edge points inside each sub-window, and use these numbers as features. The regularity of the building region and the irregularity of the background also make the orientation of the gradient a good discriminator, therefore, we calculate the mean value of the orientation of the gradient inside a sub-window and use it as a feature.

The largest sub-window has size 48×48, and the smallest is of size 6×6. We design the sub-windows such that they have at least 6×6 overlap with the current image patch (a 16×16 window). By doing so, each feature contains neighborhood information to classify the current image patch. For each sub-window in the image, the mean, variance, and histogram can be calculated efficiently using integral image [16] and integral histogram [15]. Altogether, we have around 10,000 features for each image patch.

4.2 Results

The high dimensionality of the feature space in this problem prevents binary quantile regression [10, 11] from being applicable. To test the Median Boost algorithm on the building block labeling task, we used the standard normal cumulative distribution function with $h = 0.1$ as the approximation to the indicator function. From our experience, the classifier is not sensitive to the value of h as long as $h < 0.5$. We follow the suggestion from [1, 5], and fix the step size parameter at $\nu = 0.1$. In performing the third step of the Median Boost algorithm (see Algorithm 2), the simple linear regression model with only one predictor was used as weak learner for its simplicity. The Median Boost classifier was ran for 120 iterations.

A variation of the AdaBoost cascade [16], Probabilistic AdaBoost Cascade (PABC) [18], was tested with the same set of features. The cascade structure contains four AdaBoost nodes, and each node runs 120 iterations, with decision stump as weak learner. In order to control the detection rate of the cascade, we enforce that at most 1% of positive examples could be rejected at each node by adjusting the threshold parameter in AdaBoost. As a comparison, we also tested the median classifier defined in [7] on this problem.

The first four features picked by the Median Boost algorithm are: the sum of the 12th and the 22nd features from [12], the variance of the Gabor filter response inside the sub-window at the relative location (-2, -10, 26, 26) to the top-left corner of the current image patch, the sum of the 14th and 25th features from [12], and the mean of the original image inside the sub-window at the relative location (-2, 0, 5, 5). The first four features selected by the first AdaBoost node are: Variance of the Gabor filter response inside the sub-window at the relative location (-9, -16, 26, 26), the sum of the 1st and 21st features from [12], the difference of the 2nd and 17th features from [12], and the average number of vertical edge points in the sub-window at the relative location (-16, -9, 19, 26). The lists of selected features show that Median Boost and AdaBoost agree that the features from [12] and the variance features of filter response in a sub-window are informative.

Performance Measures	PABC 1	PABC 2	PABC 3	PABC 4	median clf [7]	median boost
Detection Rate	94.27%	89.01%	83.33%	77.56%	79.49%	75.35%
False Positive Rate	25.37%	16.18%	11.58%	8.37%	22.73%	9.97%
False Negative Rate	5.73%	10.99%	16.67%	22.44%	20.51%	24.65%
Site-wise Error Rate	22.84%	15.52%	12.24%	10.18%	22.45%	11.86%

Table 1: The numerical evaluation result on 129 testing images: "PABC *n*" stands for Probabilistic AdaBoost Cascade with *n* AdaBoost nodes. The results of PABC are given in [18]

Table 1 presents the performance measures for different models. We can see that initially, with only one AdaBoost node, the detection rate of PABC is very high, this is expected because we enforce the high true positive for each node. With more AdaBoost nodes, the detection rate of PABC decreases, but the false positive rate also decreases, as a result, the site-wise classification error rate decreases monotonically. With 120 iterations, the performance of the proposed Median Boost has similar detection rate and error rate with PABC with 4 AdaBoost nodes, and this shows the efficiency of the proposed algorithm. We also observe that the median classifier in [7] has very high false positive although it has a relatively high detection rate, as a result, it has a significantly higher site-wise error rate than both Median Boost and PABC.

Fig. 1 shows the detection results on four testing images. As (b) shows, with only one AdaBoost node, PABC can detect almost all the building blocks, i.e., it has high detection rate and high false positive rate. With more AdaBoost nodes, PABC can remove some false positives, as seen from (c) to (e). (g) presents the results obtained by the proposed Median Boost, and we see the results are comparable to the results in (e), but visually better than the results in (b) to (d). (f) is the result obtained by the median classifier in [7], which shows significantly more false positives than both (e) and (g). These visual observations are consistent with the numerical measures presented in Table 1. From Fig. 1, we see the results from Median Boost and Probabilistic AdaBoost cascade are visually quite close to the ground truth which are shown in (h). Keep in mind that in order to achieve similar performance, PABC uses 4 AdaBoost nodes, totally 480 features, while Median Boost only selects 120 features.

It is worth mentioning that in conducting the experiment, we notice that the manually labeled ground truth is not consistent. For example, in some images, the building roof is labeled as positive example, while in other images, it is labeled as negative; in some images, part of the building reflection is labeled as positive while other part of the reflection is labeled as negative (see column 3 in Fig. 1). This inconsistency indicates that there are outliers in the data, and the satisfactory performance of Median Boost verifies the stability of the median information to the outliers.

5 Conclusions and Future Work

This paper proposes designing classifiers based on estimating median of binary response. Inspired by the idea of gradient boosting [5], we designed a practical algorithm which works

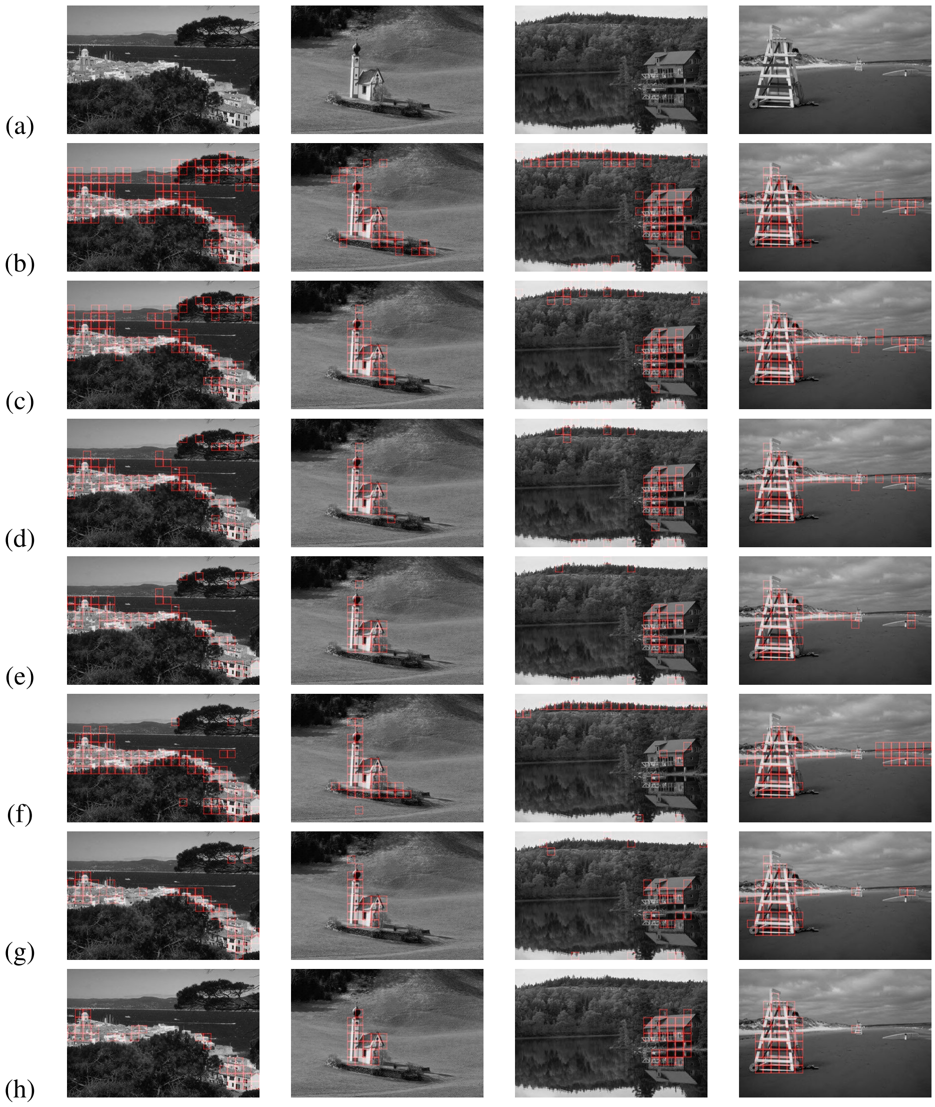

Figure 1: The experimental result on image patch labeling, the detected building patches are marked with red boundary: (a) shows the input image; the labeling results by PABC [18] with 1 to 4 AdaBoost nodes are shown in (b) to (e), respectively; (f) shows the labeling results by the Median Classifier [7] and the labeling results by the proposed algorithm are given in (g); (h) is the the manually labeled result. Please view in color for better visual effect.

in the framework of additive model, and updates the fitted model in the gradient direction by a small step in each iteration. The proposed Median Boost algorithm obtains informative features and the classifier simultaneously. On the problem of labeling building blocks in natural images, compared to the Probabilistic AdaBoost cascade, Median Boost has similar performance with far fewer iterations. Compared to another median based classifier, Median Boost performs significantly better.

In computer vision applications, we often face multi-class situation. Therefore, we plan to develop the multi-class version of Median Boost. Many existing techniques can transform multi-class task into a series of two-class tasks, for example, one-vs-all [4] and Error-Correcting Output Codes [3]. Similar to AdaBoost cascade [16, 18], it is also possible to build a median classifier cascade, and this is another future research project. To our best knowledge, this work is the first attempt to use median information for classification problems in computer vision, and we are actively seeking for other applications of the proposed Median Boost algorithm.

Acknowledgment

This work was partially supported by a 2011 Summer Faculty Fellowship awarded by Missouri State University.

References

[1] P. Bühlmann and T. Hothorn. Boosting Algorithms: Regularization, Prediction and Model Fitting. *Statistical Science*, 22, pp. 516-522, 2007.

[2] P. Bühlmann and B. Yu. Boosting with the L_2 Loss: Regression and Classification. *Journal of the American Statistical Association*, 98, pp. 324-340, 2003.

[3] T. G. Dietterich and G. Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, vol. 2: pp. 263-286, 1995.

[4] Y. Freund and R. Schapire. A Decision-Theoretic Generalization of On-line Learning and An Application to Boosting. *Journal of Computer and System Sciences*, 55(1): pp. 119-139, 1997.

[5] J. H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, Vol. 29, pp. 1189-1232, 2001.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive Logistic Regression: a Statistical View of Boosting. *Annals of Statistics*, 28, pp. 337-407, 2000.

[7] P. Hall, D. M. Titterington, and J. H. Xue. Median-based Classifiers for High-Dimensional Data. *Journal of the American Statistical Association* , vol. 104, no. 488, pp. 1597-1608, 2009.

[8] D. R. Hunter and K. Lange. Quantile Regression via an MM Algorithm. *Journal of Computational & Graphical Statistics*, vol. 9, no. 1, pp. 60-77, 2000.

[9] R. Koenker. *Quantile Regression*. Cambridge University Press, 2005.

[10] G. Kordas. Credit Scoring Using Binary Quantile Regression. In *Statistical Data Analysis Based on the L_1-Norm and Related Methods*, 2002.

[11] G. Kordas. Smoothed Binary Regression Quantiles. *Journal of Applied Econometrics*, vol. 21, no. 3, pp. 387-407, 2006.

[12] S. Kumar and M. Hebert. Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[13] S. Kumar and M. Hebert. Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2003.

[14] E. Osuna, R. Freund, and F. Girosit. Training Support Vector Machines: an Application to Face Detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 130-136, 1997.

[15] F. Porikli. Integral Histogram: a Fast Way to Extract Histograms in Cartesian Spaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[16] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec. 2001.

[17] G. R. Walsh. *Methods of Optimization*. John Wiley and Sons, 1975.

[18] S. Zheng. Probabilistic Cascade Random Fields for Man-made Structure Detection. In *Proceedings of the 9th Asian Conference on Computer Vision (ACCV)*, pp. 596 – 607, Xi'an, China, Sept., 2009.

[19] S. C. Zhu, Y. N. Wu, and D. B. Mumford. Filters, Random field And Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling. *International Journal of Computer Vision*, vol. 27, no. 2, pp.1-20, 1998.