Probabilistic Cascade Random Fields for Man-Made Structure Detection

Songfeng Zheng

Department of Mathematics
Missouri State University, Springfield, MO 65897, USA
SongfengZheng@MissouriState.edu

Abstract. This paper develops the probabilistic version of cascade algorithm, specifically, Probabilistic AdaBoost Cascade (PABC). The proposed PABC algorithm is further employed to learn the association potential in the Discriminative Random Fields (DRF) model, resulting the Probabilistic Cascade Random Fields (PCRF) model. PCRF model enjoys the advantage of incorporating far more informative features than the conventional DRF model. Moreover, compared to the original DRF model, PCRF is less sensitive to the class imbalance problem. The proposed PABC and PCRF were applied to the task of man-made structure detection. We compared the performance of PABC with different settings, the performance of the original DRF model and that of PCRF. Detailed numerical analysis demonstrated that PABC improves the performance with more AdaBoost nodes, and the interaction potential in PCRF further improves the performance significantly.

1 Introduction

Traditional pattern classification methods assume that the class labels are independent to each other. However, in real life data (e.g. sequences, images, videos), the labels of the adjacent data points are correlated. This suggests us take account of the label dependencies in designing classifiers for real life data. For example, Markov Random Fields (MRF) [6], Conditional Random Fields (CRF) [4], and Discriminative Random Fields (DRF) [9], improve the performance of an i.i.d. classification technique by taking into account the spatial dependencies.

In this paper, we are primarily interested in classifying elements (pixels or regions) of a two-dimensional image. Let $\mathbf{X}$ be the observed data from an input image, where $\mathbf{X} = \{\mathbf{x}_i\}_{i \in S}$ with $\mathbf{x}_i$ being the data from the i^{th} image site, and S is the set of all the image sites. Let the corresponding labels for the image be $\mathbf{Y} = \{y_i\}_{i \in S}$, where y_i is the label for image site i.

MRF is usually used in the generative model framework which models the joint distribution of the observed data and the labels. The posterior of the labels given the data can be expressed by Bayes' rule as

$$P(\mathbf{Y}|\mathbf{X}) \propto P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X}|\mathbf{Y})P(\mathbf{Y}). \tag{1}$$

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part II, LNCS 5995, pp. 596–607, 2010.

The prior distribution of the labels, $P(\mathbf{Y})$, is modelled as MRF. However, the likelihood term, $P(\mathbf{X}|\mathbf{Y})$, is usually very complicated, and it is a distribution in a high-dimensional space (since the image data $\mathbf{X}$ is of high dimension). Thus, it is usually very difficult, if not impossible, to find a good model for $P(\mathbf{X}|\mathbf{Y})$.

On the other hand, CRF and DRF are employed in the discriminative model framework, in which we directly model the posterior distribution of the labels given the data, $P(\mathbf{Y}|\mathbf{X})$. CRF was proposed in the context of segmentation and labelling of 1D sequences, and DRF is generalized version of CRF for 2D image data.

There are two components in DRF model, namely, the association potential and the interaction potential (see Section 2 for details about DRF model). The association potential models the local evidence which ignores the neighborhood information. In [9], the association potential was modelled by a logistic regression classifier, which can only incorporate a limited number of features, leading to restricted classification capability.

AdaBoost [2] is a classification framework which has appealing theoretical properties, and has shown impressive empirical results in a wide variety of tasks, for example, face detection [15,16,17]. This paper takes the advantage of the power of AdaBoost to incorporate more informative features for learning the association potential in DRF, thus overcoming the limitations of logistic regression model in [9]. In the learning stage, we face the problem of unbalanced training set, i.e. far less positive examples than negative examples. AdaBoost cascade [15,16,17] and WaldBoost [13] are usually used to solve this problem. However, the aforementioned methods give a results in $\{-1, 1\}$, while we need a real number for the association potential, which is the logarithm of a probability value as in [9]. To achieve this purpose, we develop Probabilistic version of AdaBoost Cascade (PABC), which calculates the posterior probability of class label when a testing example is presented. PABC is employed to learn the association potential in DRF model, and the interaction potential is learned in the same way as in the original DRF model [9]. The resulting model, Probabilistic Cascade Random Fields (PCRF), enjoys the capability of incorporating far more informative features and a more powerful association potential than the conventional DRF model.

The proposed PCRF was applied to man-made structure detection problem. We compared the performance of PABC with different settings, the performance of the original DRF model and the performance of PCRF. Detailed quantitative measures demonstrate that with more AdaBoost nodes, the overall performance of PABC improves, and with the information from interaction potential, PCRF further removes some false positives and fills in some missing parts of the object.

2 Review of Discriminative Random Fields

Discriminative Random Fields (DRF) model [9] avoids the independence assumption and seek to model the conditional joint distribution of the labels if the data is given, i.e., $P(\mathbf{Y}|\mathbf{X})$. DRF model defines the conditional probability of the labels $\mathbf{Y}$ as:

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z}\exp\left(\sum_{i\in S} A_i(y_i,\mathbf{X}) + \sum_{i\in S}\sum_{j\in\mathcal{N}_i} I_{ij}(y_i,y_j,\mathbf{X}))\right), \tag{2}$$

where A_i is the association potential that models the dependencies between the observations and the class labels, and I_{ij} is the interaction potential which models the dependencies between the labels of the adjacent elements (and the observations), and $\mathcal{N}_i$ is a neighborhood of image site i. In this paper, we assume the random field is homogeneous and isotropic, i.e., the functional forms of A_i and I_{ij} are independent of the locations i and j, hence we can simplify the notations as A and I, respectively. This model alleviates the need to model the observation data $P(\mathbf{X}|\mathbf{Y})$ (a necessary step in Bayesian statistics), and it also allows the use of arbitrary attributes of the observations without explicitly modelling them.

The association potential $A(y_i,\mathbf{X})$ reflects the local evidence of the label for image site i. For two-class classification, $y_i \in \{1,-1\}$, the association potential is modelled as [9]:

$$A(y_i,\mathbf{X}) = \log\left(\sigma(y_i\mathbf{w}^T\mathbf{h}_i(\mathbf{X}))\right), \tag{3}$$

where $\sigma(\cdot)$ is logistic regression function:

$$\sigma(y_i\mathbf{w}^T\mathbf{h}_i(\mathbf{X})) = P(y_i|\mathbf{X}) = \frac{1}{1+\exp\left(-y_i\mathbf{w}^T\mathbf{h}_i(\mathbf{X})\right)}. \tag{4}$$

In Eqn. (3) and (4), $\mathbf{h}_i(\mathbf{X})$ is the feature vector extracted from the image data for site i, and $\mathbf{w}$ is the weight vector. In principle, the feature vector $\mathbf{h}_i(\mathbf{X})$ can be any transformation of the image data. The association potential defined in Eqn. (3) makes DRF equivalent to a logistic regression classifier if the interaction potential is set to zero.

To model the interaction potential, let $\boldsymbol{\mu}_{ij}(\mathbf{X})$ be the pairwise feature vector extracted from the image data $\mathbf{X}$ which reflects the property of the image site pair (i,j). Similar to Eqn. (3), the pairwise discriminative term (a probability) is defined as

$$P(y_i,y_j|\mathbf{X}) = \sigma\left(y_iy_j\mathbf{v}^T\boldsymbol{\mu}_{ij}(\mathbf{X})\right), \tag{5}$$

where $\mathbf{v}$ is the parameter vector. The interaction potential is modelled as a convex combination of two terms, i.e.:

$$I(y_i,y_j,\mathbf{X}) = \beta\left\{Ky_iy_j + (1-K)\left[2\sigma\left(y_iy_j\mathbf{v}^T\boldsymbol{\mu}_{ij}(\mathbf{X})\right)-1\right]\right\}, \tag{6}$$

where $0 \le K \le 1$. When $K=1$, the interaction potential boils down to the Ising model, therefore the interaction potential can be thought of as a generalization of the Ising model.

Note that both the association potential $A(y_i,\mathbf{X})$ and the interaction potential $I(y_i,y_j,\mathbf{X})$ depend on the whole image $\mathbf{X}$, not only on the image data at site i or site j. This is different from the traditional classification setting.

The parameters $\boldsymbol{\theta} = \{\mathbf{w}, \mathbf{v}, \beta, K\}$ can be obtained by maximizing the pseudo-likelihood function:

$$\hat{\boldsymbol{\theta}} \approx \arg\max_{\boldsymbol{\theta}} \prod_{m=1}^{M} \prod_{i \in S} P(y_i^m | y_{\mathcal{N}_i}^m, \mathbf{X}, \boldsymbol{\theta}), \tag{7}$$

where m indexes the training images and M is the total number of training images, and

$$P(y_i | y_{\mathcal{N}_i}, \mathbf{X}, \boldsymbol{\theta}) = \frac{1}{Z_i} \exp\left\{ A(y_i, \mathbf{X}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{X}) \right\}, \tag{8}$$

with Z_i as the normalization factor. The pseudo-likelihood function given in Eqn. (7) can be maximized by linear search method [9], Newton's method, or stochastic gradient method.

3 Probabilistic AdaBoost Cascade

As a classification algorithm, AdaBoost [2] combines a set of weak classifiers (features) to form a strong classifier, and the obtained strong classifier is

$$H(\mathbf{x}) = \text{sign}\left\{ \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right\} \in \{-1, 1\}, \tag{9}$$

where $\mathbf{x}$ is the input example, $h_t(\mathbf{x}) \in \{-1, 1\}$ is the weak classifier (feature) selected at the t^{th} iteration with weight α_t, and T is the total number of iterations. It is well-known that there is a deep relation between AdaBoost and the additive logistic regression model [3], i.e.,

$$p(y|\mathbf{x}) = \frac{\exp\left\{ y \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right\}}{\exp\left\{ \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right\} + \exp\left\{ -\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}) \right\}}, \quad \text{with} \quad y \in \{-1, 1\}. \tag{10}$$

In applications, we usually have a limited number of positive examples, but abundant negative examples, that is, the training set is highly unbalanced. AdaBoost cascade [15,16], and its variant, Boosting Chain [17], are successfully used for object detection with unbalanced training set. WaldBoost [13], using sequential likelihood ratio test in decision making, implicitly builds cascade structure after every weak classifier is added. In AdaBoost cascade, before training each AdaBoost node, we can bootstrap negative examples in case there are not enough negative examples, as shown in Fig. 1. A testing example will be classified as positive if it can pass all the AdaBoost nodes; otherwise, it will be classified as a negative example.

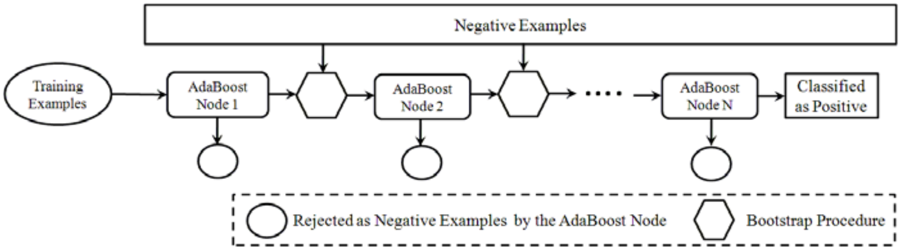


Fig. 1. AdaBoost Cascade which can bootstrap negative examples at each stage

AdaBoost cascade, Boosting chain, and WaldBoost output a value in $\{-1, 1\}$, while in certain application scenarios, we prefer a probability value $P(y = 1|\mathbf{x})$. As such, we develop Probabilistic AdaBoost Cascade (PABC) which has the same structure as the non-probabilistic version, see Fig. 1. The difference is that for each testing example $\mathbf{x}$, PABC outputs the probability $P(y = 1|\mathbf{x})$ instead of ± 1.

The training process of PABC is very similar to that of AdaBoost cascade as shown in Fig. 1, except that we use Eqn. (10) to calculate the probability value when we split the training set. Ideally, we would like to keep all the positive examples in training the AdaBoost nodes, but inevitably we will make some mistakes when splitting the training set. To keep as many positive examples as possible, we put restriction on the false negative rate for each split by the AdaBoost nodes. For each split, we also calculate the proportion of the positive examples among the rejected examples. The detailed training process is given in Fig. 2.

Let $\mathbf{x}$ be a testing example, and y be the corresponding label. We regard each of the subset rejected by the AdaBoost node $\mathcal{S}_{\mathrm{RJ}}$ as a classifier as well, and it outputs the probability

$$P(y = 1|\mathbf{x}) = P(y = 1|\mathcal{S}_{\mathrm{RJ}}, \mathbf{x}) = p_n, \tag{11}$$

that is, the proportion of positive examples in the subset $\mathcal{S}_{\mathrm{RJ}}$. From Fig. 3, it is easy to write out the posterior probability of y given the testing example $\mathbf{x}$ as:

$$P(y = 1|\mathbf{x}) = \sum_{y_1 \in \{-1,1\}} P(y = 1|y_1, \mathbf{x}) P(y_1|\mathbf{x}), \tag{12}$$

and similarly, we have the following recursive formula:

$$\begin{aligned} P(y = 1|y_{n-1} = \cdots = y_1 = 1, \mathbf{x}) = \sum_{y_n \in \{-1,1\}} & P(y = 1|y_n, y_{n-1} = \cdots = y_1 = 1, \mathbf{x}) \\ & \times P(y_n|y_{n-1} = \cdots = y_1 = 1, \mathbf{x}), \end{aligned} \tag{13}$$

where $y_i \in \{-1, 1\}$, $P(y_n|y_{n-1} = \cdots = y_1 = 1, \mathbf{x})$ is the probability calculated by the n^{th} AdaBoost node using Eqn. (10); at a terminal node, $P(y = 1|y_n, y_{n-1} = \cdots = y_1 = 1, \mathbf{x})$ is the output by the terminal node using Eqn. (11); if the

- **Input:** An initial training set $\mathcal{S} = \{(\mathbf{x}_i, y_i) : i \in \Lambda\}$, where $\mathbf{x}_i$'s are feature vectors, $y_i \in \{-1, 1\}$ is the label, and Λ indexes all the training examples. We also have a set $\mathcal{S}_{\text{neg}}$ which contains a large number of negative examples. The desired false negative rate for each cascade split fn is also given.
- For $n = 1, \cdots, N$
  1. Train a strong classifier by AdaBoost for node n using the current training set $\mathcal{S}$.
  2. For each $\mathbf{x}_i \in \mathcal{S}$, calculate the probability $P(y_i = 1|\mathbf{x}_i)$ using Eqn. (10).
  3. Split the training set $\mathcal{S}$ into $\mathcal{S}_{\text{RJ}}$ and $\mathcal{S}_{\text{go}}$, where $\mathcal{S}_{\text{RJ}}$ and $\mathcal{S}_{\text{go}}$ are the subset classified by the current AdaBoost node as negative and positive examples, respectively; when making this split, we adjust the threshold such that the false negative rate is at least as small as the given value fn; for $\mathcal{S}_{\text{RJ}}$, calculate the proportion of positive examples p_n.
  4. If there are not enough negative examples in $\mathcal{S}_{\text{go}}$, bootstrap negative examples from the given set $\mathcal{S}_{\text{neg}}$; let $\mathcal{S} = \mathcal{S}_{\text{go}}$.
- End For

Fig. 2. Training process of PABC with N AdaBoost nodes

current node is not a terminal node, then $P(y = 1|y_n, y_{n-1} = \cdots = y_1 = 1, \mathbf{x})$ is calculated recursively by Eqn. (13). Using Eqn. (12) and the recursive relation (13), we can calculate the output probability by PABC. Thus in the testing stage, PABC integrates information from every node to make decision.

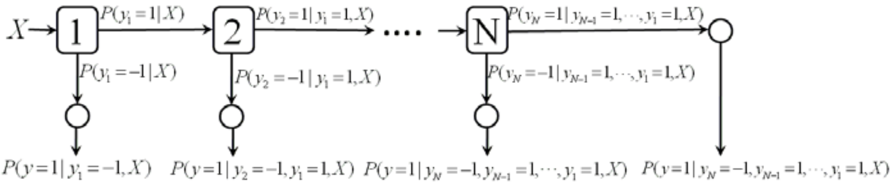

X
1
2
N
....
$P(y_1 = 1|X)$
$P(y_2 = 1|y_1 = 1, X)$
$P(y_N = 1|y_{N-1} = 1, \cdots, y_1 = 1, X)$
$P(y_1 = -1|X)$
$P(y_2 = -1|y_1 = 1, X)$
$P(y_N = -1|y_{N-1} = 1, \cdots, y_1 = 1, X)$
$P(y = 1|y_1 = -1, X)$
$P(y = 1|y_2 = -1, y_1 = 1, X)$
$P(y = 1|y_N = -1, y_{N-1} = 1, \cdots, y_1 = 1, X)$
$P(y = 1|y_N = -1, y_{N-1} = 1, \cdots, y_1 = 1, X)$

Fig. 3. Testing procedure of PABC. The boxes are the classifiers learned by AdaBoost, and the circles are the terminal nodes of the cascade. y_i is the decision result of the i^{th} AdaBoost node, and y is the output result. The probabilities are calculated according to each AdaBoost node (Eqn. 10) or from the proportions at the terminal nodes (Eqn. 11).

Tu [14] proposed a Probabilistic Boosting Tree (PBT) algorithm, of which PABC is a special case, since the chain structure in PABC is a special case of the tree structure in PBT. However, as the depth of the tree increases, the number of nodes in PBT increases exponentially, which will need much more time in the training stage than PABC. Furthermore, a tree is much more complicated than a chain, thus PBT is more likely to over-fit the data than PABC. In [18], a learning procedure called Probabilistic Boosting Network (PBN) is presented, which is implemented by means of an efficient graph structure. In [18], PBN was used to classify object and estimate pose parameters at the same time, while in this paper, we are only focus on classification. In PBN, if there is no pose parameter, the graph structure of PBN will boils down to the structure of AdaBoost cascade.

4 Probabilistic Cascade Random Fields

The original DRF model [9] learns the association potential by a logistic regression model. However, the logistic regression model can only incorporate a small number of features, and the classification capability of logistic regression model is not strong. Moreover, logistic regression often does not estimate appropriate parameters, and this is especially true for image data where feature vectors may have a high number of dimensions and possibly there are high degree of correlations among features.

Fortunately, the DRF framework allows a flexible choice of the association potential. By making use of the strong classification ability of Support Vector Machines (SVM), Lee et al. [5] proposed to use probabilistic version of SVM [11] for learning the association potential. Although SVM has good classification performance, it needs a large amount of training time when the feature number and training set are large. More over, SVM does not have an explicit solution to the problem of imbalanced training set which is common in applications.

This motivates us to apply the introduced PABC algorithm to learn the association potential since PABC can deal with a large number of features and a large number of training examples. Due to the powerful feature selection mechanism of AdaBoost, PABC will not select correlated features. Furthermore, PABC is designed for imbalanced data, thus it is less sensitive to imbalanced training set compared to SVM and AdaBoost.

The learned association potential by PABC algorithm is expressed as

$$A(y_i, \mathbf{X}) = \log P(y_i = 1|\mathbf{X}), \tag{14}$$

where $P(y_i = 1|\mathbf{X})$ is fitted by the procedure described in Fig. 2, and calculated for a given sample by Eqn. (12) and Eqn. (13).

This work still adopts the interaction potential as in Eqn. (6), also see [9]. We maximize the pseudo-likelihood function to estimate the parameters $\boldsymbol{\theta} = (\mathbf{v}, \beta, K)$ in the interaction potential, i.e.

$$(\hat{\mathbf{v}}, \hat{\beta}, \hat{K}) \approx \underset{(\mathbf{v},\beta,K)}{\arg\max} \prod_{m=1}^{M} \prod_{i \in S} P(y_i^m | y_{\mathcal{N}_i}^m, \mathbf{X}, \boldsymbol{\theta}). \tag{15}$$

To ensure that the log-likelihood is convex and prevent over-smoothing due to the pseudo-likelihood approximation, we assume a Gaussian prior on $\mathbf{v}$ and use the penalized log pseudo-likelihood function [10]

$$l(\mathbf{v}, \beta, K) = \sum_{m=1}^{M} \sum_{i \in S} \left\{ A(y_i, \mathbf{X}) + \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{X}) - \log Z_i \right\} - \frac{1}{2}\mathbf{v}^T\mathbf{v}, \tag{16}$$

where $I(y_i, y_j, \mathbf{X})$ depends on the parameters $(\mathbf{v}, \beta, K)$ as defined in Eqn. (6), and Z_i is a normalization constant which also depends on the parameters $(\mathbf{v}, \beta, K)$. Note that $A(y_i, \mathbf{X})$ is learned by PABC and calculated according to Eqn. (14), therefore in the optimization procedure $A(y_i, \mathbf{X})$ can be ignored since they are constants to $(\mathbf{v}, \beta, K)$. We use gradient ascent to maximize the penalized log pseudo-likelihood function in Eqn. (16).

Given a testing image $\mathbf{X}$, our goal is to find the most probable label configuration $\mathbf{Y}^*$ for $\mathbf{X}$, i.e., solve a Maximum A Posteriori (MAP) problem:

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}), \tag{17}$$

where the probability $P(\mathbf{Y}|\mathbf{X})$ is evaluated according to Eqn. (2) with the learned parameters. Since the probability distribution only contains unary and binary terms, the MAP can be solved by max-flow/min-cut type of algorithms [7]. As in [9], we choose to use iterated conditional modes (ICM) [1] for inference due to its simplicity, and it yields a local maximum of the posterior probability. Given an initial labelling, ICM iteratively maximizes the local conditional probability, that is, for each image site, we update the label by

$$y_i \leftarrow \arg\max_{y_i \in \{-1,1\}} P(y_i|\mathbf{Y}_{\mathcal{N}_i}, \mathbf{X}). \tag{18}$$

5 Experiment

We test the proposed model on the task of man-made structure detection from natural images. 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 blocks, and each image block is an image site in our model. The ground truth was generated by manually labelling every image site as *building* and *non-building*. There are 5,203 building blocks and 36,269 non-building blocks in the training set, and 6,372 building blocks and 43,164 non-building blocks in the testing set[1].

5.1 Features

For the man-made structure detection problem, we use the features described in [8,9] as our first set of features, which are based on the weighted histogram of the gradient orientation. Please refer to [8] for more details. We also use different combinations (sum, difference, etc.) of features from [8].

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

[1] The original image data and the labels are provided by [9].

For each sub-window, we can also calculate the histograms from each filter response, and use each bin of the histogram as a feature, and the entropy of the histogram can be used as a feature as well to evaluate the regularity of the sub-window. We also notice that man-made structures 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 map, 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 must have at least 6×6 intersection with the current image site (a 16×16 window). By doing this, each feature contains neighborhood information to classify the current image site. This feature design strategy is consistent with our notation $P(y_i|\mathbf{X})$, i.e. the class label for image site i depends on the whole image, not only $\mathbf{x}_i$ itself. For each sub-window in the image, the mean, variance, and histogram can be calculated efficiently using integral image [15] and integral histogram [12]. Altogether, we have around 10,000 features for learning the association potential.

In learning the association potential by PABC, the first 4 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) to the top-left corner of the current image site, with error rate 0.189; the sum of the first and 21st features from [8], with error rate 0.298; the difference of the second and 17th features from [8], with error rate 0.349; the average number of vertical edge points in the sub-window at the relative location (-16, -9, 19, 26) to the top-left corner of the current image site, with error rate 0.386.

To learn the interaction potential, we use features $\boldsymbol{\mu}_{ij}(\mathbf{X})$ as those used in [9], that is, the difference of two vectors from [8] at image sites i and j, such that the feature vector $\boldsymbol{\mu}_{ij}(\mathbf{X})$ encodes the difference between image sites i and j.

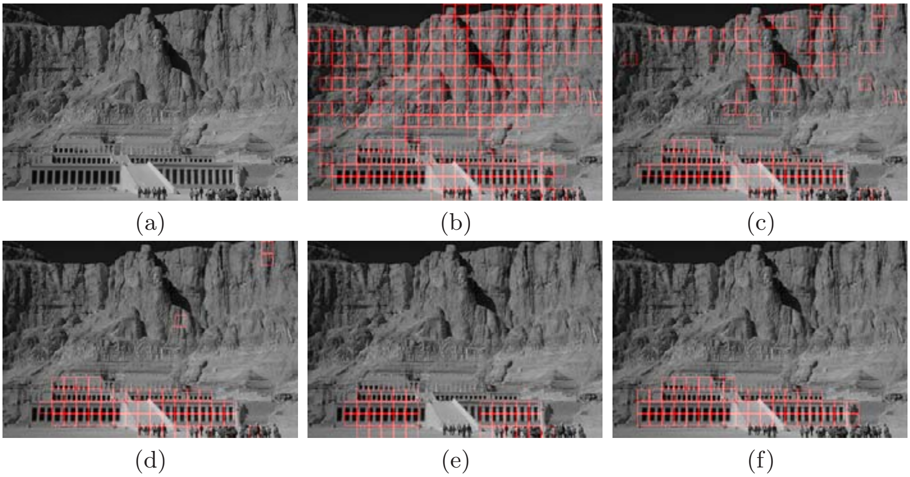
5.2 Results

When applying PABC[2] to learn the association potential for PCRF, we use 5 AdaBoost nodes, and for each of them, we select 120 features. We feed all the positive examples to the first AdaBoost node, and each AdaBoost is learned with 10,000 negative examples. When splitting the training set, the false negative rate is set to be 0.015, and we bootstrap negative examples if necessary. Learning the association potential by PABC needs about 2 hours, and learning the interaction potential by maximizing pseudo-likelihood needs about 5 minutes with 40 iterations to converge. In the testing stage, for each input image, the computer needs about 20 seconds to output the detection result. The computer has a 2GHz CPU and 3.25G Bytes memory.

[2] The PABC is implemented based on the source code provided by [16].

From Eqn. (2), when the interaction potential is set to zero, the PCRF model is reduced to a classification model learned by PABC. In this case, given the observed image data $\mathbf{X}$, the optimal label configuration $\mathbf{Y}^*$ is found by maximizing the class posterior. In another word, the optimal label for the i^{th} site is

$$y_i^* = \underset{y_i \in \{-1,1\}}{\arg\max}\, P(y_i|\mathbf{X}). \tag{19}$$


(a) (b) (c)

(d) (e) (f)

Fig. 4. The experimental result on man-made structure detection, the detected building blocks are marked in red boundary: (a) shows the input image in gray scale; (b) and (c) are the detection results from PABC with 1 and 5 AdaBoost nodes, respectively; (d) and (e) are the detection results from PCRF and the original DRF, respectively; (f) is the the manually labelled result. Please view in color for better visual effect.

Fig. 4 shows the detection result on a testing image. As can be seen from (b), initially, with one AdaBoost node, PABC can detect almost all the building blocks, i.e., it has high detection rate, but it also has high false positive rate. With more AdaBoost nodes, PABC can remove some false positives, as seen from (c). (d) is the result obtained by PCRF, which shows that the interaction potential further removes the false positives, although there are still false positives compared to the manually labelled result in (f). (e) shows the result obtained by the original DRF model[3], from which we can see that the original DRF model has fewer false positives, but it has more false negatives.

Table 1 presents the performance measures of the model with different settings. As we can see, with more AdaBoost nodes, the detection rate decreases, but the false positive rate also decreases, as a result, the site-wise classification error rate decreases monotonically. This is expected because PABC aims

[3] The MATLAB toolbox of DRF model for man-made structure detection was downloaded from http://www.cs.ubc.ca/∼murphyk/Software/CRF/crf.html

at minimizing the error rate. The table shows that with the information from the interaction potential, the PCRF improves the detection rate slightly, but the false positive rate drops significantly.

Table 1. The numerical evaluation result on 129 testing images: PABC n stands for PABC with n AdaBoost nodes

Performance Measures	PABC 1	PABC 2	PABC 3	PABC 4	PABC 5	PCRF
Detection Rate	94.27%	89.01%	83.33%	77.56%	72.18%	72.64%
False Positive Rate	25.37%	16.18%	11.58%	8.37%	6.23%	3.94%
Site-wise Error Rate	22.84%	15.52%	12.24%	10.18%	9.01%	6.95%

Our final result has better detection rate than that reported in [9], but slightly worse false positive rate. The reason is that in [9], the parameters for the association potential and for the interaction potential are estimated simultaneously, while the PCRF model learns the model parameters separately, which might be a suboptimal strategy. Pursuing learning methods which can estimate the parameters simultaneously needs more investigation.

6 Conclusions and Future Works

This paper develops the probabilistic version of AdaBoost cascade (PABC), which outputs a probability value instead of -1/1 value. We use PABC to learn the association potential in the DRF model, resulting the Probabilistic Cascade Random Fields (PCRF) model. We applied the proposed model to the task of man-made structure detection, and compared the performance of PABC with different settings, the performance of the original DRF model, and the performance of PCRF. Detailed qualitative and quantitative analysis showed that PABC improves the overall performance with more AdaBoost nodes. With the information from interaction potential, PCRF further removes some false positives and fills in some missing parts of the object of interest. Our final result is comparable to that reported literature.

In this paper, only the association potential is learned by PABC, while the interaction potential is learned by a simple logistic regression model. Therefore, the current PCRF model still has limited ability to combine more informative features in the interaction potential. Our next step is using PABC to learn the interaction potential. Also, it is desirable to test the proposed approach to other applications and compare to state-of-the-art results, e.g. face detection [15].

Acknowledgement

The author would like to thank the anonymous reviewers for their comments which improve the quality of this paper. Part of this work was supported by CNAS Research Fellowship of Missouri State University.

References

1. Besag, J.: On the Statistical Analysis of Dirty Pictures. J. of Royal Statisticial Society B-48, 259–302 (1986)
2. Freund, Y., Schapire, R.E.: A Decision-theoretic Generalization of on-line Learning and an Application to Boosting. J. of Computer and Sys. Sci. 55 (1997)
3. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. Annals of Statistics 28 (2000)
4. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proc. of ICML (2001)
5. Lee, C., Greiner, R., Schmidt, M.: Support Vector Random Fields for Spatial Classification. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 121–132. Springer, Heidelberg (2005)
6. Li, S.Z.: Markov Random Field Modeling in Image Analysis. Springer, Tokyo (2001)
7. Kolmogorov, V., Zabih, R.: What Energy Functions can be Minimized via Graph Cuts? IEEE Trans. on PAMI 26(2), 147–159 (2004)
8. Kumar, S., Hebert, M.: Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field. In: Proc. of CVPR (2003)
9. Kumar, S., Hebert, M.: Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification. In: Proc. of ICCV (2003)
10. Kumar, S., Hebert, M.: Discriminative Fields for Modeling Spatial Dependencies in Natural Images. In: Proc. of NIPS (2004)
11. Platt, J.C.: Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In: Advances in Large Margin Classifiers, pp. 61–74 (1999)
12. Porikli, F.: Integral Histogram: a Fast Way to Extract Histograms in Cartesian Spaces. In: Proc. of CVPR (2005)
13. Šochman, J., Matas, J.: WaldBoost – Learning for Time Constrained Sequential Detection. In: Proc. of CVPR (2005)
14. Tu, Z.: Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. In: Proc. of ICCV (October 2005)
15. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: Proc. of CVPR (December 2001)
16. Wu, J., Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Fast Asymmetric Learning for Cascade Face Detection. IEEE Trans. on PAMI 30(3), 369–382 (2008)
17. Xiao, R., Zhu, L., Zhang, H.: Boosting Chain Learning for Object Detection. In: Proc. of ICCV (October 2003)
18. Zhang, J., Zhou, S.K., McMillan, L., Comaniciu, D.: Joint Real-time Object Detection and Pose Estimation Using Probabilistic Boosting Network. In: Proc. of CVPR (2007)