Boosting part-sense multi-feature learners toward effective object detection

Shi Chen, Jinqiao Wang*, Yi Ouyang, Bo Wang, Changsheng Xu, Hanqing Lu

Institute of Automation, Chinese Academy of Sciences, Beijing, China

1. Introduction

In recent years, statistical learning methods are mainly employed to achieve high accuracy for object detection systems. These statistical learning methods have witnessed rapid development and obtained excellent performance in some applications. For example, neural network-based face detectors were proposed in [1,2]. A face classifier was learned by support vector machines in [3,4]. In these approaches, object detection problems were cast into selecting and combining features to a learner or classifier from the training samples. While designing such a single and distinct learner up-to-down is very difficult; AdaBoost [5] provides a simple and effective approach to construct a final classifier by selecting several “weak learners” at each iteration and boosting them into a more distinct classifier by linear combination. To cope with visual face detection task, Viola and Jones [6] designed a learner as a decision stump (a decision tree with only two leaf nodes) and each weak learner is usually associated with one individual visual feature, e.g. Haar-like feature, as shown in Fig. 1a.

Despite of the successful applications in face detection, we also noticed that the single-feature weak classifiers are sometimes “too weak” in some other complex detection problems (e.g. human detection). This similar situation will happen in later stages of the boosting process, it becomes more and more difficult to separate the samples as the training process continued. The experimental result reported in [6] showed that the error rates of single-feature learner increase dramatically from 0.1–0.3 to 0.4–0.5. These weak classifiers chosen in later stages perform just a little better than random guess, and can hardly help to decrease the training error and lead to very long training phase.

Hence, a lot of approaches are proposed to improve the discrimination ability of learners to decrease the training time. An intuitive idea is to combine several multiple decision stumps to a middle classifier instead of using decision stumps as learner at each iteration of the boosting process, as shown in Fig. 1b. In [7,8], different forms of combination have been designed to joint or composite multiple learners to each learner. However, there are two major problems of these methods. Firstly, when the features are binarized into decision stumps, much descriptive information is lost. Some correlations with features are also lost by simple threshold. Secondly, besides the combination forms, the number of features involved in the learners also plays an important role in the learners. More features in each learner can improve the discrimination ability of learners, but if excessive features are introduced into the learners, the learners would be prone to over-fitting and poorer generalization performance in testing phase. Therefore, we should get a balance between the discrimination ability and generalization of learners. Nevertheless, in [7,8], the number of features involved in each learner is obtained by heuristically rules or fixed in advance experimentally.

In this paper, we propose an effective learner design and implementation framework to further improve the object detection performance in the framework of boosting learning. Generally, our
approach can be divided into two modules. The first module is the learner design mechanism of our approach. We put forward a novel part-sense multi-feature learner (PML) to select and linearly combine several multiple features from the feature pool, as shown in Fig. 1c. Additionally, we formulate the solution algorithm of our learner design as a weighted LASSO regression problem. Through LARS method [9], our approach can choose few features self-adaptively and efficiently for each learner. The other module of our method is the $L_1$-regularied gradient boosting algorithm, with which we can integrate our part-sense sparse features learner into gradient boosting framework to obtain an object classifier. The framework of our method is shown in Fig. 2. The advantages of our approach can be concluded in three aspects:

Firstly, instead of conventional combining decision stumps to each weak learner, we select and combine multiple local features linearly to form higher accuracy multi-feature learners at each iteration. The proposed method can achieve more discriminative power for each weak learner in a natural way than these stump-based methods. In addition, considering that most useful correlation exists in the local structure and to decrease the computation complexity, we design our learner in part sense which means that we only select and combine features in the local neighborhood, which will remarkably decrease the computation complexity without loss of discriminative performance.

Secondly, the feature selection and combination is formulated as a weighted LASSO regression problem. Through the mature LARS method for LASSO regression [10], we can trade off the discrimination and generalization ability with the number of features in contrast to heuristic or experimental strategies used in previous methods.

Finally, we demonstrate that the feature selection and combination process can be naturally fitted in the gradient boosting [11] framework. In fact, we can interpret the weighted LASSO regression as finding the steepest-descent direction with sparsity in function space. With new $L_1$-regularied gradient boosting algorithm, we can combine our part-sense multi-feature learners to the final classifier as simply and effectively as original AdaBoost algorithm.

The following of this paper is organized as follows. At first, related work is discussed in Section 2. Section 3 describes our motivation and design of proposed part-sense multi-feature learner (PML). In Section 4, the weighted LASSO regression and the solution algorithm of our learner design are presented in detail. $L_1$-regularied boosting algorithm, which integrates our part-sense sparse features learner into gradient boosting framework, is introduced in Section 5. The experimental results on face detection and human detection tasks are reported in Section 6. We conclude the paper with future work in Section 7.

2. Related work

Concentrating on the design and implementation of learners in the boosting framework, we only introduce the related work according to the learner design modes.

2.1. The single-feature learner

As the pioneer work of applying AdaBoost to object detection, Viola and Jones [6] constructed weak learners pool $h_i(\cdot)$ by associating each of them with an individual Haar-like feature $x_i$, and the learner is simply designed as a decision stump

$$h_i(x_i, \beta) = \begin{cases} 1, & \text{if } p \cdot h(x_i) > p \cdot \beta_i, \\ 0, & \text{otherwise,} \end{cases}$$

where $\beta_i$ is a threshold and $p$ is the parity indicating the direction of the inequality sign.

However, single-feature weak classifiers are not always distinctive enough for complex vision tasks, and ‘too weak’ classifiers cause a long training phrase. To overcome this problem, combining decision stumps is adopted in the existing literature [7,8] to improve the discrimination of the weak learners.

![Fig. 1. Three weak learner design mechanisms in boosting algorithm: (a) traditional single-feature learner, (b) multi-stump learner and (c) multi-feature learner.](image1)

![Fig. 2. Boosting part-sense multi-feature learners based object detection framework.](image2)
2.2. The multi-stump learner

The model of combination multiple decision stumps, shown in Fig. 1b, can be summarized as follows:

\[ h_i(x; \beta) = f(h_{i1}(x_1; \beta_1^i), h_{i2}(x_2; \beta_2^i), \ldots, h_{ik}(x_k; \beta_k^i)), \]

(2)

where \( f(.) \) is the combination function. The difference of these approaches can be distinguished by these various combination functions.

In [7], the combination function is represented as joint learner by integrating the binary output computed from prefixed number \( F \) of weak learners. Firstly, through binarization, each feature value can be represented by a binary variable. Hence, the feature space is divided into \( 2^N \) sub-cubs by binary coding. The classification result of the joint classifier in each sub-cub is obtained based on the weighted voting in the feature space. Based on the Bayes decision rule, the corresponding weak classifier is constructed as Eq. (3)

\[ h_i(x) = \begin{cases} +1, & \text{if } P(y = +1|x_j) > P(y = -1|x_j), \\ -1, & \text{otherwise}, \end{cases} \quad \forall x \in j, \]

(3)

where \( j \) is the index of the sub-cubs, and \( P(y = +1|x) \) and \( P(y = -1|x) \) are evaluated with respect to weights \( G_i(x) \) of examples.

It is a critical problem to determine \( F \) for the Joint Haar-like feature. In [7], \( F \) is selected by hand with an empirical upper bound or chosen through the cross-validation. Furthermore, though the best feature combination can be found by exhaustive search from all possible feature combinations, it is not feasible for a limited training time. Thus, the approximate solution for efficient feature selection has to be adopted. In [7], Sequential Forward Floating Selection [12] is utilized to search for a sequential sub-optimal solution.

In [8], a data mining-driven approach, named as frequent item set mining (FIM) [13], is introduced to mine the compositional learner for boosting. In [8], through taking the compositional decision stumps as feature itemset and defining the frequency of an itemset, frequent itemset mining (FIM), an efficient data mining technique for pattern classification is adopted to composite multiple features to learners. However, it is difficult to find out the optimal classification rule or compositional learner. Hence, several heuristic and empirical parameters (support \( \lambda_1 \) and confidence \( \lambda_2 \)) are designed to evaluate the quality of the learners. In experiments, \( \lambda_1 \) and \( \lambda_2 \) are also prefixed in advance.

In [14], Huang proposed an enhanced biologically inspired model (EBIM) which imposes sparsity constraints and applies a feedback procedure that selects effective features for combination. It achieved good results on object recognition street scene database. However, the sparsity constraints and feedback procedure are both based on the original BIM. Therefore, it will be not easy to use EBIM to select and combine general features, such as Haar features.

In addition, it should be noticed that in existing approaches, features and learners are not distinguished strictly because each feature is firstly quantized to a corresponding binary learner. Therefore, though the existing work also claimed to combine or join features, the approaches actually combined or joined learners.

3. Part-sense multi-feature learner

We consider two aspects to design a part-sense multi-feature learner.

Firstly, we linearly combine several features directly to our part-sense multi-feature learner as shown in Fig. 1c

\[ h_i(x; \beta) = \sum_{m=1}^{F} \beta_m x^m. \]

(4)

Compared with the combination of decision stumps, our approach can learn a more distinct learner naturally. As shown in Fig. 3, it is a true instance in face detection with Haar-feature. Specifically, two best individual features can only obtain the classification precision of about 60% by traditional method [6], which combines decision stumps. As each stump is only constructed by single feature, the classifier plane is a horizontal or vertical line as shown by black lines in Fig. 3a. Different decision stump combination methods actually make the horizontal or vertical line shift. However, when combining two features with \( 5_f1 + 9_f2 \), we can get a much better classification precision of about 80% with threshold 0.4. The classifier plane by our approach is also shown in Fig. 3b by sideling lines. Obviously, the description ability of our classifier is stronger than decision stumps combination methods. For this instance, we can see that the relation between the features could further improve the discriminative ability of each learner, while it would be lost if the features are cut to decision stumps in advance.

Secondly, in real-world application, it's easy to access large number of feature sets. For the instance in face detection, the total number of four basic types of Haar-like features in a single 20 \( \times \) 20 image is as much as 78,460. It is time and memory consumption to get best features in such huge whole feature set. Hence, considering that most useful correlations exist in the local structure, we design a part-sense learner to automatically select and combine features in the local neighborhood. The experimental results show that though part of feature set is considered, the performance of learners is hardly influenced.

In particular, we first select the best discriminative features from the whole feature pool. Then centered on this feature, we only consider the features which are neighboring to this seed-feature in

(a) Combine multiple decision stumps.

(b) Combine multi-feature.

Fig. 3. Comparison of two multi-feature learner design approaches.
local image regions as candidate features set to construct the multi-feature learners. Through a weighted LASSO regression model, we can get few features in candidate set for our part-sense multi-feature learners efficiently. Compared with traditional combination approaches under the global perspective, our method is much faster and obtains almost the same discriminative ability as the global result. The flowchart of constructing part-sense multi-feature learner is illustrated in Fig. 4.

4. Weighted LASSO regression for sparsity of features

When combining local features for a more discriminative weak learner, we also desire that only a small portion of features are selected. In this way, we can take the classification ability and sparsity simultaneously into consideration. The detail algorithm as follows.

Firstly, we utilize LASSO [10] as the feature selection method to find the optimal combination of features as follows:

\[
\beta_t = \arg \min_{\beta} \sum_{i=1}^{M} \left( y_i - \sum_{m=1}^{F} \beta_m x_i^m \right)^2 + C \| \beta \|_1,
\]

where \( y_i \) is the label of the sample \( i \) and \( \beta_m, x_m \) stand for the mth feature and the corresponding coefficient.

The reason why we adopt LASSO is that L1 penalty is added into the LASSO model besides regularizing the Least Squares. Compared with L2, the major advantage of \( L_1 \) minimizations is the sparsity of resolution. As revealed by Tibshirani [10] and Wright [15], if the constraint that number of features is far more than the number of samples is satisfied, the solution of the \( L_1 \)-minimization problem is equal to the solution to the \( L_0 \) minimization problem which can be interpreted as the sparsest solution. This constraint is naturally satisfied in the tasks of object detection with local features (the number of object samples is \( \sim 10^k \), the number of local features is \( \sim 650 \) k). However, the resolution of \( L_2 \) is dense. The sparse resolution means we only need several features to do classification, which will accelerate both training and detection process. The computation of the lasso solutions is a quadratic programming problem [10] and can be tackled by standard numerical analysis algorithms. In our work we adopt another method, Least Angle Regression (LARS) [9], because it is a useful and less greedy version of traditional forward selection methods, which fits “least angel direction” to the residual error. Furthermore, it exploits the special structure of the lasso problem, and provides an efficient way to compute the solutions simultaneously for all values of \( L_1 \) penalties. Details for LARS can be found in [9].

We improve the standard LASSO regression to adopt the boosting framework, named Weighted LASSO Regression. Given the weights \( D(i) \) corresponding to each sample \( i \), the weighted LASSO regression is to minimize the flowing problem:

\[
\beta_t = \arg \min_{\beta} \sum_{i=1}^{N} \left[ y_i D(i) - \sum_{m=1}^{F} \beta_m x_i^m \right]^2 + C \| \beta \|_1.
\]

In this way, we can easily obtain the best multiple features for each learner in boosting with different distributions of the samples. In addition, \( C \) is the tuning parameter to trade off the discriminative and generalization abilities. When \( C \) is smaller, it achieves the learners with stronger discrimination, while \( C \) is bigger, it leads the less elements in \( \beta_t \) would be nonzero, which encourages the sparsity and the generalization power of learners at the expense of their discriminative ability.

In addition, it should be noticed that there are some methods trying to find sparse learners for the final classifier from the perspective of boosting learning. In [16], Xi added \( L_1 \) regularization to the total exponential loss to form an \( L_1 \) regularized loss minimization problem (RLMP) and transfer it to a convex minimization problem in a small set of the learners that has been added before. In [17], Zhang et al. added \( L_1 \) regularization to the generalized loss function in additive model learning. However, solving these \( L_1 \)-regularized generalized loss function is not trivial. Complex search method is referred to get an approximately optimal result. Basically, these approaches do not conflict or disagree with our approach to find sparsity of features in each learner, which means we can use two methods simultaneously, first seek the sparsity of features and then pursue the sparsity of learners.

5. \( L_1 \)-regularized gradient boosting

5.1. Preliminaries of AdaBoost and gradient boosting

AdaBoost, like many other classification models can be written as a linear combination of some simpler classifiers or learners methods or so called additive models:

\[
F_T(x) = \sum_{t=1}^{T} \alpha_t f_t(x; \beta_t),
\]

where \( x \) is the input data, \( f_t(x; \beta_t) \) are any arbitrary learner of \( x \) with parameter \( \beta_t \). Typically, the parameters and classifiers \( \alpha_t, f_t(x; \beta_t) \) are estimated by minimizing some loss function \( \Psi \), which measures the prediction errors over training data \( \{x_m,y_m\} \):

\[
\{x^*, f^*_c(x; \beta_c)\} = \arg \min_{\{f(x; \beta)\}_{t=1}^{N}} \Psi \left( y_m, \sum_{t=1}^{N} \alpha_t f_t(x; \beta_t) \right).
\]

Directly optimizing such loss function is often difficult. However, this optimizing in additive models can be done efficiently by forward stagewise strategy. The basic idea is, sequentially adding new classifiers and parameter \( f_t(x; \beta_t), \alpha_t \) to the final classifier.
without changing the parameters that have been added before. This forward stagewise method can be written as following loops:

1. Forward stagewise search: For \( t = 1, 2, \ldots, n \):
   \[
   \{ x_t, f_t(x; \beta_t) \} = \arg \min_{\{ h_t, f_t(x) \}} \sum_{i=1}^{N} \Psi(y_i, F_{t-1}(x) + x_T f_t(x; \beta_t)).
   \]
   (9)

2. Update the additive models:
   \[
   F_{t}(X) = F_{t-1}(X) + ax_T f_t(x; \beta_t).
   \]
   (10)

In [17], Friedman proved that Adaboost can be interpreted as an additive model by forward stagewise approach with the loss function \( \Psi \) is the exponential loss:

\[
\Psi(y, f(x)) = \exp\left(-y f(x) \right).
\]  
(11)

The gradient boosting [11], from the perspective of numerical optimization in the function space, interpreted traditional stagewise additive expansion as the steepest-descent minimization. In detail, Instead of obtaining the solution by searching Eq. (9) in function space, gradient boosting firstly calculate the best steepest-descent step \( g_T(X) \) at the current approximation \( F_{t-1}(X) \):

\[
T \quad F_{t}(X) = E_{y_i} \left[ \frac{\partial \Psi(y_i, F(X))}{\partial F(X)} \right]_{F(X) = F_{t-1}(X)}.
\]  
(12)

Then, to choose the member of the parameterized class that is most parallel in the data space with negative gradient \( \{-g_T(x)\}^N_{i=1} \) by least-squares minimization:

\[
f_t(x; \alpha_t) = \arg \min_{f_t(x; \alpha_t)} \sum_{i=1}^{N} \left( g_T(x) - \beta f_t(x; \alpha_t) \right)^2.
\]  
(13)

Finally, the optimal step along the direction is obtained by a linear search. Finally, the \( f(x; \alpha_t) \) is added to the optimal model with the step \( \beta_t \).

Specifically, the steepest-descent direction \( g_T(x) \) of the \( i \)th sample at iteration \( t \) in the data space is calculated as:

\[
g_T(x_i) = \left[ \frac{\partial \Psi(y_i, F(X))}{\partial F(X)} \right]_{F(X) = F_{t-1}} = y_i \exp\left(-y_i F_{t-1}(x_i) \right).
\]  
(14)

Where \( y_i \) is the label of the \( i \)th sample. We can rewrite Eq. (13) to connect the direction representation to the traditional weight-updating strategy. Similar to Real AdaBoost [19], we initialize the weight of each sample \( D(i) \) as \( 1/N \) and update it at each iteration according to:

\[
D_{t+1}(i) = \frac{D_t(i) \exp\left(-a_T y_i h(x; \beta_t) \right)}{Z_{t}},
\]  
(15)

where \( Z_t \) is a normalizer. Combining (14) and (15), it is easy to obtain:

\[
\frac{g_T(x_i)}{\beta_t} = \sum_{i=1}^{N} \left[ y_i \exp\left(-y_i F_{t-1}(x_i) \right) \right] = N y_T D_{t-1}(i) \prod_{t=1}^{T-1} z_t.
\]  
(16)

As we only concern the direction of \( g_T(x_i) \), the scale of \( g_T(x_i) \) can be removed as:

\[
g_T(x_i) = y T D_{t-1}(i).
\]  
(17)

5.2. The proposed \( L_1 \)-regularized gradient boosting

In contrast to selecting the best weak learner which is most parallel with the steepest-descent direction, we proposed a method combines local features to construct the optimal direction with sparsity.

In the canonical gradient boosting, it chooses the member of the parameterized direction which is most parallel in the \( N \)-dimensional data space with negative gradient \( \{-g_T(x)\}^N_{i=1} \) by least-squares minimization. However, in this paper, we approach the steepest direction \( g_T(x) \) by combining several features using the least square loss:

\[
\beta_t = \arg \min_{\beta} \sum_{i=1}^{N} \left[ y_i D_{t-1}(i) - \gamma \sum_{m=1}^{F} \beta T x_m \right]^2.
\]  
(18)

Taking the sparsity into consideration, the optimal combination can be formulated as:

\[
\beta_t = \arg \min_{\beta} \sum_{i=1}^{N} \left[ y_i D_{t-1}(i) - \gamma \sum_{m=1}^{F} \beta T x_m \right]^2 + C \| \beta \|_1,
\]  
(19)

Eq. (19) is similar to the weighted LASSO regression problem (Eq. (6)) expects optimal scale \( \gamma \). However, the scale information is useless in this fitting steepest direction process because we will calculate the optimal step \( \alpha_T \) specifically in the next step. More detailed explanation can be found in [11]. Hence, we can interpret fitting steepest direction with sparsity of feature as weighted LASSO regression, where \( C \) is the trade-off between the fitting-precision and sparsity. Though SVM can also be adopted to solve Eq. (18), the sparsity of features is not considered in traditional SVM. Moreover, the purpose of SVM is to maximize the margin, while LARS, adopted in our approach, aims to fit the steepest direction with sparsity. Hence, compared with the SVM method, our approach is more suitable for the framework of boosting, especially, the gradient boosting.

Once obtained the steepest direction \( \beta_t \) at each iteration, the optimal step \( \alpha_T \) can be obtained by linear search following the original gradient boosting algorithm. In our work, we calculate it analytically in a similar way as [19], which greedily minimizes the

\[
h(x) = \frac{0.42^*}{0.57^*} + \frac{0.33^*}{2.0}.
\]  

Fig. 5. Comparison of gradient boosting and \( L_1 \)-regularized gradient boosting. Left: gradient boosting and right: \( L_1 \)-regularized gradient boosting.
bound 2 of training error on each round of boosting. We can apply this idea in the choice of $x_t$.

Let $p = \sum_{i=1}^{f} |\beta_i|^1$, and $u_i = y_i h_t(x_i; \beta_i) \xi y_i \beta_i x_i$. Since each feature $x$ is normalized to $[-1, +1]$, $u \in [-p, +p]$. Then the loss at $r$ iteration is

$$\text{Loss}(F_t(x_i)) = \sum_{i=1}^{N} W_i \Psi(y_i, F_t(x_i) + x_i h(x_i; \beta_i)).$$  \hspace{1cm} (20)

Minimizing Eq. (20) gives $x_t$:

$$x_t = \ln((p + r)/(p - r))/2p.$$ \hspace{1cm} (21)

where $r = \sum_{i=1}^{N} D_t(i) u_i$. The detail proof and the convergence of training error upperbound are proved in Appendix. The whole learning algorithm is shown in Algorithm 1.

Algorithm 1  \hspace{1cm} L1-regularized gradient boosting

1 Calculate the optimal features $f_i$ with respect to the current sample weights.
2 Obtain candidate feature set $F$.
3 Obtain the part-sense multi-feature learner $h(x, \beta_i)$ by weighted LASSO regression Eq. (6).
4 Let $\sum_{i=1}^{r} |\beta_i|_1$, calculate training error $r$ of $h(x, \beta_i)$ with respect to the current sample weights:

$$r = \sum_{i=1}^{N} D_t(i) y_i h(x; \beta_i)$$

5 Compute the optimal weight (step) of the new weak learner:

$$\alpha_t = \ln((p + r)/(p - r))/2p.$$  \hspace{1cm} (6)

6 Update:

$$D_{t+1}(i) = D_{t}(i) \exp\left(-y_i \beta_i h(x; \beta_i) \right) / z_t,$$

where $z_t$ is the normalization factor.

End

Output the final hypothesis:

$$H(x) = \text{sign} \left( \sum_{i=1}^{T} \alpha_t M_t \beta_i \xi x_i \right)$$

where $\beta_i$ is the index of nonzero element in $\beta_i$, $M_t$ is the number of nonzero in $\beta_i$ at $M_t$ iteration.

The difference between gradient boosting and our $L_1$-regularized gradient boosting can be illustrated in Fig. 5.

In a special case, only one weak learner is selected, according to LARS algorithm, the best feature would be chosen. Then it can be seen that our algorithm is equal to the traditional AdaBoost. Therefore, our algorithm can be regarded as a natural generalization of the tradition AdaBoost and performs at least as good as the traditional one in theory.

6. Experiments

To evaluate the performance of the proposed approach, we performed the thorough experiments on face and human dataset. We compare the proposed method (denoted as PML) with two representative approaches, the AdaBoost [7] with single-feature-learners (SFL) and the combination of multiple decision stumps (the joint learners [7], denoted as JL). For the sake of fairness, the three methods are compared based on the number of features rather than the number of iteration or the number of weak learners. All three methods stop when the number of selected features reaches 1000 on face dataset and 700 on Human dataset.

6.1. Evaluation results on face dataset

In this evaluation, we setup the experiments similar as [7] to compare the performance of learner design. The positive dataset consists of 10,000 frontal face images, which are collected from FERET [20], CMU [21] and Yale [22]. In the matter of collection of negative samples, we adopted the methods in [7], not the classic random cropping in nonface images [6]. Specifically, we first collected 1000 images.

The negative samples are very similar to the positive samples obtained by our methods from the web that do not contain any face. Then we train a classifier based on our approach and a classifier by Mita et al. [7] using all the positive samples and 10,000 negative samples cropped by random. Finally, we use these two classifiers to classify the nonface images and collect 10,000 negative samples which is misclassified by both two classifiers. The typical training samples are shown in Fig. 6. All samples are the same size of 20 × 20. In our experiments, the 10,000 positive samples are divided into a training set of 7000 samples and a testing set of 3000 samples. The negative testing set of 3000 negative samples is collected by random cropping in nonface images. The classic Haar-like features [6] are chosen for all methods in the experiment. We fix the number of joint learners as three which is suggested as the optimal number for JL [7], and $C = 0.3$ for PML which would choose 3–4 features on average at each iteration. The comparison is conducted as the following three aspects.

6.1.1. The comparison of discriminative abilities on face dataset

Here we present the learning curves for the strong classifier composed of up to 1000 weak features, because what we aim to evaluate here is the performance of learner design methods instead of the system work of AdaBoost. Therefore, the complex cascade system is not adopted. Similar designs of experiments are in [7,23]. We firstly compare the discriminative ability of three kinds of learner construction methods. The performance is measured by detection rate (DR) and false alarm (FA) on the testing set.

PML improves the DR by 3.3% and 7.1% compared with JL and SFL when the number of the weak learner is 1000. From Fig. 7, we can observe the DR of PML performs similar as JL when the number of weak classifier is below 250. However, from Fig. 8, with the same number of weak learner, our method always achieves lower FA than the two others. By combining 1000 weak classifiers, the false alarm of our approach is 3.4% versus 7.3% of the joint learner, and 9.9% of single-feature-learners.

From Figs. 7 and 8, the following conclusions can be drawn: (1) given the same number of features, our proposed learners achieves a better DR and FA than JL and SFL and (2) our approach needs fewer weak features than original AdaBoost and joint learner methods in order to achieve the same false alarms. For example, the lowest test error 9.9% of Original AdaBoost with 1000 weak features can be achieved by our method with only about 260 features.

In addition, we must point out that through bootstrapping process, our negative samples are really very hard to distinguish. The detection rate of best learners is less than 0.7, while in traditional random cropping method, the detection rate of best learners always can achieve as high as 0.85.

6.1.2. Evaluation the part-sense model on face dataset

In this section, we compare the performance of part-sense model and global model. In contrast to part-sense model which
combine features in the local neighborhood, global model are allowed to combine features everywhere. With the huge number of features, we cannot calculate the global optimal solution directly. The similar problem is also issued.
in [24], which tried to use $L_1$-regularization to select features for SVM in face detection. Following [24], we adopted the re-sampling method to estimate the optimal result of the global model. In detail, we build 200 feature subsets each with random 10% of the whole feature set and use LARS to selected 60 features in each feature subset. Then, a secondary feature set is produced by combining all selected features in each feature subset. The secondary set consists of 12,000 features and this feature set can be regarded as the representative of the whole feature set. Therefore, solution on the secondary feature set is an approximation to that of the global model.

The discriminative abilities of two methods are compared in first five iterations. The size of part-sense region is fixed as $8 \times 8$, which contains about 12,000 features as candidate feature set. Fig. 9 shows features selected in first three learners. We can see that the learners have find out some correlation exists in the local structure. Especially, the first learner obviously discovers the correlation between the features round right eye of face. Table 1 shows the effectiveness of the part-sense model. Though only part features are taken into considered, the detection rate of training dataset is almost as good as the global method. The difference between two methods is just from 0.2% to 1.9%. Nevertheless, the computation complexity of our method is greatly reduced than the global one.

6.1.3. Tuning the penalty $C$ on face dataset

To illustrate the influence of penalty $C$ for the generalization error in our approach, we perform experiments on the testing data for four times with different $C$, which range from 0.1 to 0.4. We use equal error rate defined in [23] to measure the performance, and all experiments are stopped whenever 1000 features have been selected, as shown in Fig. 10.

![Fig. 9. The first three learners constructed by our approach on face dataset.](image)

### Table 1

Comparison of the training error of two methods in first five iterations on face dataset.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Features</td>
<td>Training error</td>
<td># Features</td>
<td>Training error</td>
<td># Features</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.308</td>
<td>3</td>
<td>0.319</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.326</td>
<td>4</td>
<td>0.345</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.321</td>
<td>3</td>
<td>0.332</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.355</td>
<td>4</td>
<td>0.357</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0.332</td>
<td>4</td>
<td>0.348</td>
<td></td>
</tr>
</tbody>
</table>

A smaller $C$ will have more basis weak features selected, thus could rapidly reduce the training error. However, from Fig. 10 it is also found that choosing too many basis weak classifiers (When $C = 0.1$) at each iteration is prone to over-fitting. As the features increases, the generalization error does not decrease correspondingly even increase when the number of features is 145 and 200. This result is consistent with the theoretical analysis in Section 4. Experiments show that $C = 3$ is achieve the best balance of discriminative and generalization abilities for getting the lowest equal error rate.

6.2. Evaluation results of human detection

We use the INRIA pedestrian dataset [25], the MIT pedestrian dataset [26] for performance evaluation. In these datasets, training and testing samples are all of 64 $\times$ 128. The positive set consists of 5500 images, and we use 4000 images for training and the left 1500 for testing. For negative samples, we first collected 2000 non-person images from the internet, and then we divide the non-person images evenly to generate 5000 randomly training and testing samples cropped from non-person images individually. Some samples are shown in Fig. 11.

Though many novel features, e.g. HOG [25,27] and Shapelet [28], have been created for human detection, we just use the simple histogram-based image descriptors in [29], because the aim of our work is not to design novel features, but to study how to combine simple and low computational features to form a discriminative learner with generalization abilities. The result of combining histogram-based image descriptors proved that our method is a promising direction to explore the general object detection with simple feature sets.

Specifically, each output of Weighted Fisher Linear Discriminant (WFLD) is regarded as one dimension continuous feature. We compare the proposed method to classic boosted histogram approaches [29], which can be seen as single-feature learner, denoted as SFL. We also compare our method with JL as in Section 6.1. Moreover, we use the same idea of [7] to simulate the method of combining multiple decision stumps. We choose three best features as in Section 6.1.

The comparison is also conducted as the following three aspects.

6.2.1. The comparison of discriminative abilities on human dataset

Similar to face detection, the discriminative performance is also measured by the detection rate and false alarm on testing dataset. From Figs. 12 and 13, we can find out our method achieve best performance both in detection rate and false alarm rate. Comparison with face detection, the task of human detection is more difficult for all three methods achieve poor performance. On the other hand, the difference between our method and joint feature learners are very close in early stage of boosting process. However, in later stage of boosting training phase, our approach achieves distinct better performance in both detection rate and false alarm rate. The reason is that the number of features in the joint features learners is fixed to 3, which can construct learners of discriminative ability in early stages. But in later stages, just three features are not enough to construct discriminative learners in the form of joint features learner, while our approach can adaptively add the number of features to improve the discriminative ability. The statistics show the number of features in our learner increases from 3–5 to 5–10 during the training phase.

6.2.2. Evaluation the part-sense model on human dataset

We perform the experiment on human dataset similar to the face dataset. However, the feature number of histogram-based image descriptors is much bigger than that of the face detection since the size of human is 128 $\times$ 64 rather than 20 $\times$ 20. Moreover,
for the histogram-based descriptor, its discriminative ability gets better as the region gets bigger. Therefore, we cannot draw an intuitive figure to validate the correlation in local structure like Fig. 8 because the regions are too big and overlapping. However,
our part-sense model is still useful if we judge the distance between the centers of two regions whether is close enough instead of judging the regions whether is in the neighborhood. In experiments, we set the distance threshold equals to 50 pixels. The comparison result of the global method and generalized part-sense method is shown in Table 2. The difference with two methods is from 0.2% to 1.9%, which shows the effectiveness of generalized part-sense method.

### Table 2
Comparison of the training error of two methods in first five iterations on human dataset.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Global method by Destrero et al. [24]</th>
<th>Part-sense method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Features</td>
<td>Training error</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.254</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.274</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.283</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.279</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0.273</td>
</tr>
</tbody>
</table>

6.2.3. Tuning the penalty C on human dataset

The influence of penalty C to the EER in human dataset is also measured, and the results are shown in Fig. 14. The experiment is repeated for four times with different C which ranges from 0.2 to 0.8. All experiments are stopped whenever the feature number reach 700.

Similar to the results in face detection, it can be observed that a smaller $\beta$ will have more basis weak classifiers selected, which is prone to over-fitting. However, because the description capacity and the complexity of histogram-based image descriptors are much higher than Haar-like features, the phenomenon of over-fitting is prominence than the situation in face detection. When the number of feature increases from 50 to 80, the generalization error also increases from 0.35 to 0.4. On the other hand, for the ability of histogram-based image descriptors is much higher than Haar-like features, we need increase the value of C to raise the penalty of $L_1$ regularization. The best $C$ is 0.6 bigger than 0.3 in face detection.

7. Conclusions

In this paper, from the aspect of weak learner design we have proposed a novel approach towards effective object detection. Aiming at building a part-sense multi-feature learner by combination of multi-feature directly with $L_1$ regularization, we formulate the weak learner design as a weighted LASSO regression, in which its solution can be obtained directly by Linear Angel Regression (LARS). Furthermore, we design a robust $L_1$-regularized gradient boosting algorithm to combine our part-sense multi-feature learner for training object classifier. Extensive experiments and comparison with the original AdaBoost with single-feature-learners and the approaches with combination multiple decision stumps show the effectiveness of the proposed approach.

Though it is an open question in boosting literatures that which types of weak classifiers (the form and degree of complexity) should be applied for boosting, compared with previous approaches, our approach is still a more promising direction to
combine multiple features to weak learners for taking both discrimination and generalization abilities of learners into consideration.

In the future, we will further evaluate our approach on other objects, such as car, bike and horse in the more datasets, e.g. Pascal VOC.

Acknowledgments

The research is supported by National Natural Science Foundation of China (Grant Nos.: 60835002, 60905008, 90920303), and National Basic Research Program (973) of China under Contract No. 2010CB327905.

Appendix A. Appendix

We prove the Minimizing Eq. (20) gives $z_t$ to Eq. (21)

\[
\text{Loss}(F_t(x_i)) = \sum_{i=1}^{N} \Psi(y_i, F_{t-1} + z_t h_t(x_i; \beta_t)) = \sum_{i=1}^{N} \exp(- y_i(F_{t-1}(x_i) + z_t h_t(x_i; \beta_t)))
\]

\[
= \text{const} \times \sum_{i=1}^{N} D_{t-1}(i) \exp(z_t u_i) \text{by Eq. (16)}
\]

\[
= \prod_{i=1}^{T} Z_t = \prod_{i=1}^{T} \frac{Z_t}{2} \left( \frac{\beta^2 - P^2}{P} \right) \text{ where } P = \sum_{m=1}^{r} |\beta_m| \text{ and } u_i = y_i h_t(x_i; \beta_t) \text{ is as similar as [19] through derivation of the righthand of Eq. (22)}
\]

\[
\text{to minimize the Loss}(F_t(x_i)) \text{ we can easy get:}
\]

\[
z_t = \frac{(\ln(p + r) / (p - r))}{2p}
\]

where $r = \sum_{m=1}^{r} D_{t-1}(i) u_i$. Plugging into Eq. (22), This choice gives the upper bound

\[
Z_t \leq \sqrt{P^2 - r^2} / P
\]

Since $u_i \in [-p, +p]$, It is easy to obtain $Z_t \leq 1$. Hence, the error upper-bound will have a convergent result.

References