1.3 INDUCTIVE ALGORITHMS FOR CLASSIFICATION, CLUSTERIZATION, RECOGNITION

Multiple Instance Learning with Gradient Boosting Trees

Polovinkin Alexey[1], Martyanov Vladimir[1], Eruhimov Victor[2,3]

[1]Intel Corporation LTD/ASCML department, Turgeneva 30, N.Novgorod, Russia
[2] Computer Vision Center "Argus", Torgovaya 18/4, N.Novrorod, Russia
[3] N.I. Lobachevski State University of Nizhni Novgorod, Gagarina av. 23, Russia

aleksey.polovinkin@intel.com, vladimir.martyanov@intel.com, ve@argus-cv.com

Abstract. *We consider the multiple instance learning problem with application to object recognition. An approach that iteratively refines the training set and eventually gives more weight to those instances that are relevant to object description is presented. At each iteration, Gradient Boosting Trees model is built and is used to calculate sampling probabilities for the next training stage. We evaluate the proposed iterative approach on several datasets and show that it is capable of finding relevant instances and can be far superior to the base learner in terms of detection accuracy and purity.*

Keywords

computer vision, object recognition, multiple instance learning, gradient boosting trees.

1 Introduction

In this paper we consider the object recognition problem, which is currently one of the most actively researched topics in computer vision field and has applications in many domains. The common approach to this problem is to use the so-called “bag-of-features” methods [1, 2], that utilize representation of an image as an unordered collection of local descriptors. An interest point detector ([3]) finds keypoints, the number of which may differ from image to image, and feature vectors are computed as functions local regions around the keypoints (an alternative approach is to segment an image into a collection of regions and extract a descriptive feature vector from each of them).

We consider a subclass of recognition problems where one feature vector (one instance / one interest point / one sub-image) is sufficiently significant to describe the object, and all the other feature vectors are considered background. For example, if the task is to detect whether an object of a certain type is present in an image or not, we assume that after segmenting the image into regions, the object to be detected can be located in one region only.

Let us give a formal definition of multiple instance learning problem. In multiple learning problem instances are combined into bags (i.e.multisets) and these bags rather than individual instances are labeled for training. We assume that every instance in a bag labbeled negative is actually negative, whereas at least one instance in a bag labeled positive is actually positive. As stated above we consider subclass of multiple instance learning problems where exactly one instance in the positive bag is actually positive and other instances are actually negative.

2 Multiple instance learning with gradient boosting trees

The proposed multiple instance learning with gradient boosting trees (MI_GBT) algorithm operates in iterative fashion. First, we label as “positive” all instances from the „positive“ bags, and label as “negative” all instances from the “negative” bags. It should be noted that at this initial stage most of the instances labeled as “positive” are in fact negative. A classification model capable of assigning confidence scores to its predictions (such as Gradient Boosting Trees (GBT) [4]) is trained, and its predictions are used to construct the training set for the next iteration, by resampling the “positive” part of the training data with probabilities calculated from the assigned confidence levels. The output of the algorithm is "probability distribution" for each image, each value in this distribution being the likelihood that the particular feature vector from this image corresponds to the object. Of course, the model built on the final iteration can be applied to images that were not present in the training set, to select the feature vectors that most likely correspond to the object and to determine whether the object is present in the new image or not. Formal description of the algorithm is given below.

The training set consists of p positive labelled bags and n negative labelled bags, each bag $X_k, k = 1,..., p+n$ contains m_k instances $\{x_{k,1}, x_{k,2},..., x_{k,m_k}\}$. Let $h_i, i = 1,..., p$ be the unknown index of actually positive instance in the i-th bag, $p_{ij} = P(h_i = j), i = 1,..., p, j = 1,....,m_i$ - estimates of probabilities that j-th instance is actually positive in the i-th bag. Formal description of the algorithm is listed below:

MI_GBT(*numIterations*, *K*, α)

for (iter := 1; iter <= *numIterations*; iter++) do steps (1)-(4)

1) learn *K* GBT models on the training set (using subsampling).

2) calculate $f_{ijk} \in [-1;1], i = 1,..., p, j = 1,....,m_i, k = 1,..., K$ - prediction of k-th GBT model on the j-th instance into the i-th bag. Calculate average of predicted values: $f_{ij} = \frac{1}{K}\sum_{k=1}^{K} f_{ijk}, i = 1,..., p, j = 1,....,m_i$

3) estimate p_{ij} values:

$$\tilde{p}_{ij} = \frac{1}{1+e^{-\alpha f_{ij}}} \cdot \prod_{l=1, l\neq j}^{m_i} \frac{1}{1+e^{\alpha f_{il}}}, i = 1,..., p, j = 1,....,m_i ,$$

$$p_{ij} = \frac{\tilde{p}_{ij}}{\sum_{l=1}^{m_i} \tilde{p}_{il}}, i = 1,..., p, j = 1,....,m_i$$

4) create new training set:

a. instances $x_{k,1},..., x_{k,m_k}, k = p+1,..., p+n$ that describe negative bags are kept unchanged;

b. for each positive bag $X_k, k = 1,.., p$ create new multiset of instances $\tilde{x}_{k,1},..., \tilde{x}_{k,m_k}$ by random sampling from initial multiset $x_{k,1},..., x_{k,m_k}$ with probabilities p_{kj} .

3 Experimental results and analysis

The proposed approach has been tested on several synthetic and real-world datasets, consistently showing both good ability to select the significant feature vectors and superior detection performance (in terms of ROC curves) compared to straightforward application of the same underlying learner without the iterative estimation.

3.1 Artificial data set

Let each instance be a numerical vector in R^d . All components of feature vector x are sampled from uniform distribution on [0; 1]. Instance is actually positive if first *m* components lie in hypercube with punctured center area: $(x_1,..., x_m) \in [a;b]^m \setminus [a+dx; b-dx]^m$. Examples of positive feature vectors (for *m* = *d* = 2) are shown on Figure 1.

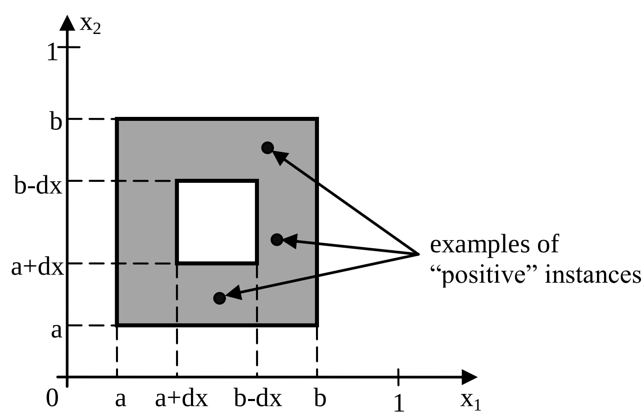

Fig. 1. Examples of "positive" instances.

Training dataset consists of 500 „positive“ bags and 500 „negative“ bags; test dataset contains 1000 bags of each kind. Number of instances in each bag is equal to 150, dimension of feature space d = 50, m = 8, a = 0.1, b = 0.2, dx = 0.025. ROC curve estimated on the test dataset is shown on Figure 2.

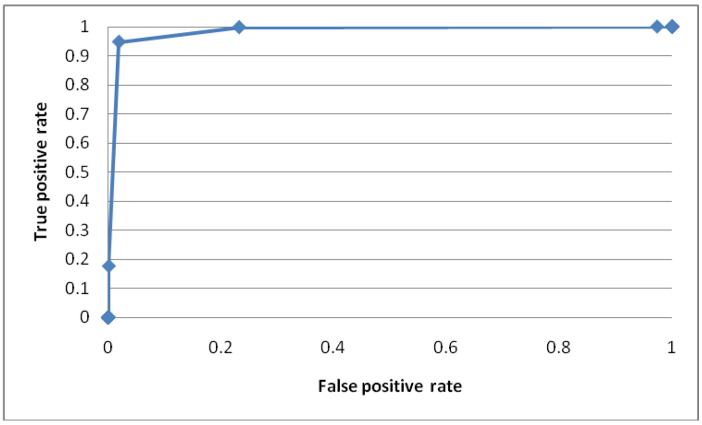

Fig. 2. ROC curve estimated on the test part of the artificial dataset.

3.2 „Digits“ data set

In this section we explore the problem of the digits detection in the image. For experimental purposes we implemented the generator of data (color(RGB) images without alpha channel). Digits (from 0 to 9) are defined by $w_{digit} \times h_{digit}$ binary masks.

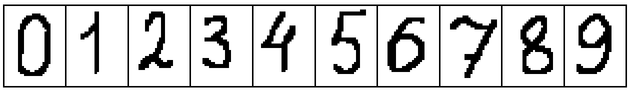

Fig. 3. Binary masks of digits.

Let w – width of the image, h – height of the image (we assume all images in the dataset of equal size). To generate background images the intensity of each pixel is uniformly sampled from [0; 255] interval. The same procedure is performed at first when we generate image containing object (digit). Then random digit (from 0 to 9) and random position of the digit in the image are selected. Intensities of pixels at the corresponding positions are changed to 0 with probability p (p is the parameter of generator). Examples of such images from train (for the different values of p) and test data sets are shown on Figures 4, 5.

To calculate feature vectors (instances) both from train and from test data sets we scan the image by the sliding window. Size of window is equal to the size of the digit mask, scanning step is equal to the half of digit mask size for each dimension. Then features are extracted from the each color channel of the segment. We use next simple features: mean, standard deviation, 9 spatial and 9 central moments of pixel intensities (so the dimension of feature space is ((2 + 9 + 9) * 3 = 60).

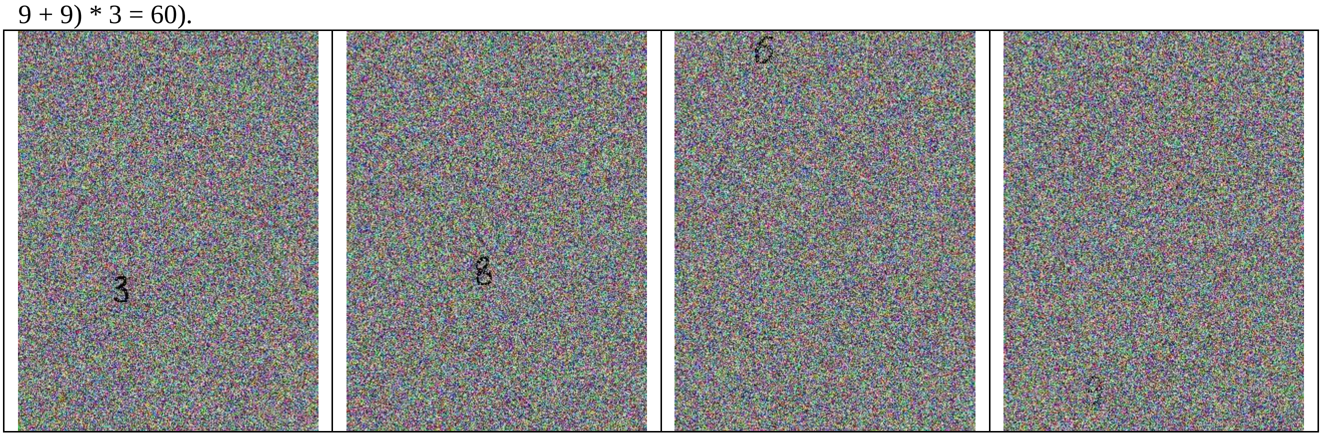

Fig. 4. Examples of generated images containing digit from train data sets (p = 0.9, p=0.7, p=0.5, p=0.3).

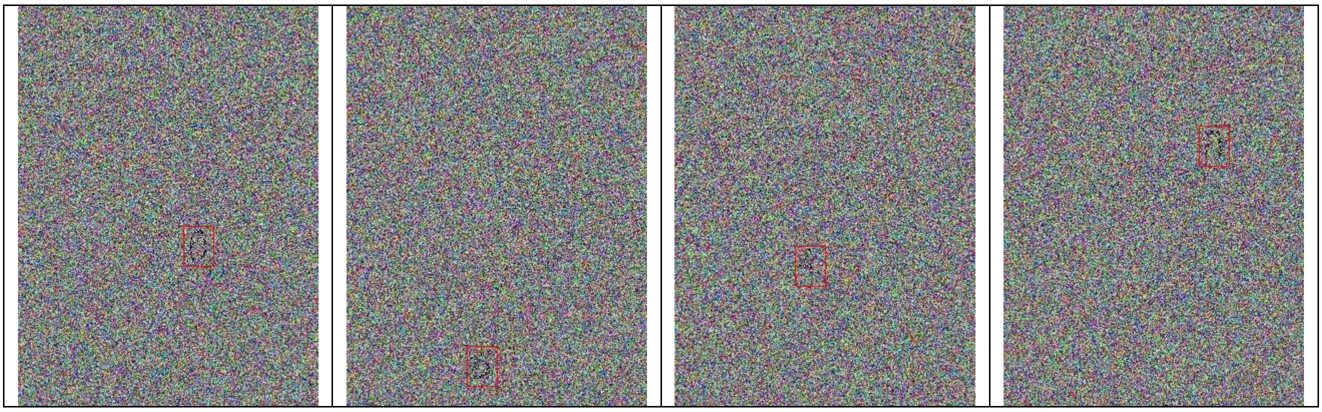

Fig. 5. Examples of images from the test data set with the estimated most probable position of the digit (p=0.3).

Train dataset consists of 100 images containing digits and 100 background images; test dataset contains 500 images of each kind. We used next parameters: w = 400, h = 300, w_{digit} = 30, h_{digit} = 40 (so the number of instances in the bag = the number of segments = 361), p = 0.3 (which corresponds to very noisy images). ROC curve estimated on the test dataset is shown on Figure 6.

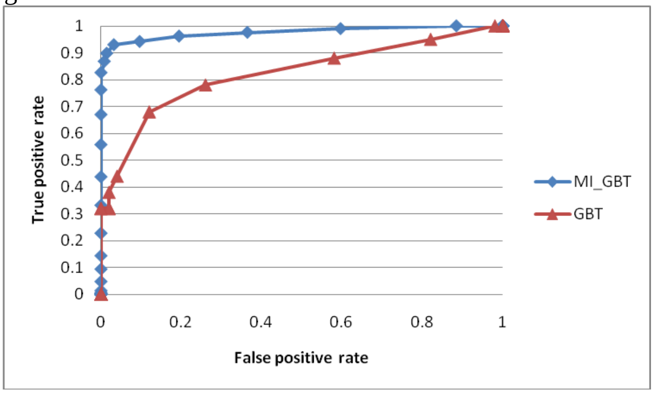

Fig. 6. ROC curve estimated on the test part of the "digits" dataset.

3.3 Algorithm analysis

We suppose that actually positive instances should be localized at the one or more certain regions of the feature space. Let us consider the dataset containing feature vectors from R^2, where all actually positive instances are located in $[0.1;0.2]\times[0.1;0.2]$, false "positive" and background instances are uniformly distributed in $[0;1]\times[0;1]\setminus[0.1;0.2]\times[0.1;0.2]$ as shown on Figure 7.

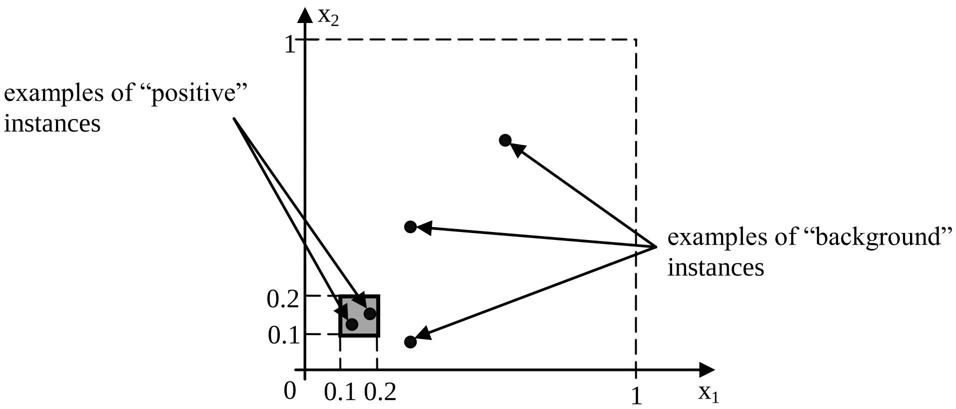

Fig. 7. Examples of "positive" and "negative" instances.

Due to the absence of “background” samples in the $[0.1;0.2]\times[0.1;0.2]$ region the values of GBT predictions f_{ijk} there are greater than in the rest of the feature space. Average values of GBT predictions after the first iteration of the algorithm are shown on Figure 8.

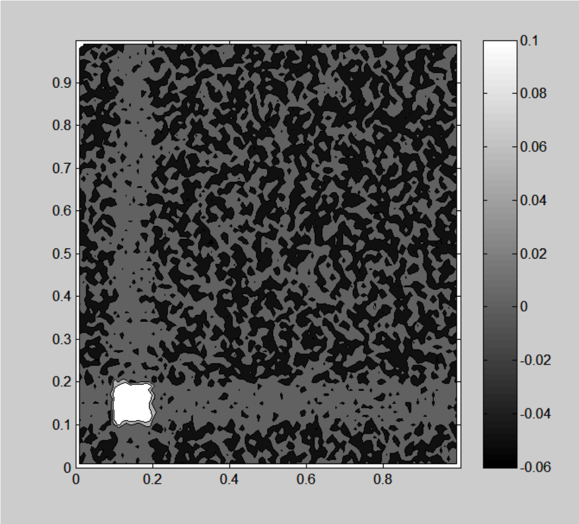

Fig. 8. Average prediction of GBT models.

So the estimates of probabilities p_{ij} (accordingly step 3 of the MI_GBT algorithm) are greater for actually positive instances than for „background“ instances labelled as „positive“. On the next iteration the fraction of actually „positive“ instances in newly sampled feature bags will be greater, so it is expected that after some iterations sampled „feature bags“ will contain a low fraction of „background“ samples labelled as „positive“ and the generalization ability of GBT models will increase.

The necessary condition for this to work is that the ratio of densities of actually positive instances and negative instances in the region of interest ($[0.1;0.2]\times[0.1;0.2]$ in our example) must be greater than the ratio of densities of actually negative instances in „positive“ bags and other negative instances elsewhere.

4 Conclusions

Iterative application of a GBT learner has shown excellent detection results on our synthetic datasets. It has much better accuracy than a straightforward application of GBT to the same input data. Additional useful property of the algorithm is that it outputs a probability distribution over individual instances, so one can estimate which of the feature vectors in the “positive” training bag (or in the test case) corresponds to the object of interest. Application of our method to real-world tasks of object detection requires domain-specific feature extraction and is a subject for future work. An obvious direction for improving this technique is generalization to the case when several training feature vectors in each bag can be positive.

5 Acknowledges

This research (work of Victor Eruhimov) was supported by the Russian Federal Program "Scientists and Educators in Russia of Innovations", contract number 02.740.11.5131

References

[1] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: An In-Depth Study. Technical report RR-5737, INRIA Rhone-Alpes. 2005

[2] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan. "Object Detection with Discriminatively Trained Part-Based Models" IEEE Pattern Analysis and Machine Intelligence (PAMI). Sept 2009.

[3] C. Schmid, R. Mohrand and C. Bauckhage, Evaluation of Interest Point Detectors, Int'l Journal of Computer Vision, 37(2), 151-172, 2000

[4] J.Friedman. Greedy function approximation: a gradient boosting machine. Technical report, Dept. of Statistics, Stanford University, 1999.