华南理工大学

South China University of Technology

The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Shen Fu
Haishan Ao

Supervisor:
Qingyao Wu

Student ID:
201720144955
201720145051

Grade:
Graduate

December 22, 2017

Face Classification Based on AdaBoost Algorithm

***Abstract*—AdaBoost is an aggressive learning algorithm which produces a strong classifier by choosing visual features in a family of simple classifiers and combining them linearly. This experiment aims to implement a face detection method using AdaBoost. Results on real world examples are presented. The detector we accomplished yields good detection rates with frontal faces.**

I. Introduction

FACE detection is a computer vision task which consists in detecting one or several human faces in an image. It is one of the first and the most important steps of Face analysis. Usually, the methods for face recognition or expression recognition assume that the human faces have been extracted from the images, but while the human visual system permit us to find instantaneously faces in our purview indifferently of the external conditions, doing the same automatically with a computer is a quite difficult task.

AdaBoost can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

This experiment post aims at providing us with intuitions towards face detection method using AdaBoost that will help us put them to use. We are first going to introduce the theory of AdaBoost. And then, we will set up AdaBoost algorithm to detect one or several human faces in different images. Finally, we will give a conclusion to summarize this experiment.

II. Methods and Theory

A. AdaBoost

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire, who won the 2003 Gdel Prize for their work. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

AdaBoost can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

Every learning algorithm tends to suit some problem types better than others, and typically has many different parameters and configurations to adjust before it achieves optimal performance on a dataset, AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier[1][2] When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative 'hardness' of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.

Pseudocode for AdaBoost is shown in Figure 1. Here we are given m labeled training examples $(x_1, y_1), ..., (x_m, y_m)$ where the $x_i^{'}$s are in some domain χ, and the labels $y_i \epsilon -1, +1$. On each round $t = 1, ..., T$, a distribution D_t is computed as in the figure over the m training examples, and a given $weaklearner$ or $weaklearningalgorithm$ is applied to find a $weakhypothesish_t : \chi \to -1, +1$, where the aim of the weak learner is to find a weak hypothesis with low weighted error ε_t relative to D_t. the $final$ or $combinedhypothesis$ H computes the sign of a weighted combination of weak hypotheses

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{1}$$

This is equivalent to saying that H is computed as a weighted majority vote of the weak hypotheses h_t where each is assigned weight α_t.

B. Normalized pixel difference feature space

The Normalized Pixel Difference (NPD) feature be- tween two pixels in an image is defined as

$$f(x, y) = \frac{x - y}{x + y} \tag{2}$$

where $x, y \geq 0$ are intensity values of the two pixels, and $f(0, 0)$ is defined as 0 when $x = y = 0$.

The NPD feature measures the relative difference between two pixel values. The sign of $f(x, y)$ indicates the ordinal relationship between the two pixels x and y, and the magnitude

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$.
- Aim: select h_t with low weighted error:

$$\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.
- Update, for $i = 1, \ldots, m$:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Fig. 1. Pseudocode for AdaBoost

of $f(x, y)$ measures the relative difference (as a percentage of the joint intensity $x + y$) between x and y. Note that the definition $f(0, 0) \triangle 0$ is reasonable because, in this case, there is no difference between the two pixels x and y. Compared to the absolute difference $|x + y|$ NPD is invariant to scale change of the pixel intensities.

The NPD feature has a number of desirable properties. First, the NPD feature is antisymmetric, so either $f(x, y)$ or $f(y, x)$ is adequate for feature representation, resulting in a reduced feature space. Therefore, in an $s \times s$ image patch (vectorized as $p \times 1$, where $p = s \cdot s$), NPD feature $f(x_i, y_j)$ for pixel pairs $1 \leq i < j \leq p$ is computed, resulting in $d = p(p-1)/2$ features. For example, in a 20×20 face template, there are $(20 \times 20) \times (20 \times 20 - 1)/2 = 79,800$ NPD features in total. We call the resulting feature space the NPD feature space, denoted as

$$\Omega_{npd} (\in R^d) \tag{3}$$

Second, the sign of $f(x, y)$ is an indicator of the ordinal relationship between x and y. Ordinal relationship has been shown to be an effective encoding for object detection and recognition because ordinal relationship encodes the intrinsic structure of an object image and it is invariant under various illumination changes. However, simply using the sign to encode the ordinal relationship is likely to be sensitive to noise when x and y have similar values. In the next section we will show how to learn robust ordinal/contrastive relationships with NPD features.

Third, the NPD feature is scale invariant, which is expected to be robust against illumination changes. This is important for image representation, since illumination change is always a troublesome issue for both object detection and recognition.

Fourth, as shown in Appendix A, the NPD feature $f(x, y)$ is bounded in [-1,1]. The bounded property makes the NPD feature amenable to histogram binning or threshold learning in tree-based classifiers.

III. Experiments

A. Dataset

The experiment uses 1000 provided pictures , in which 500 are human face RGB images. The training dataset and validation dataset include 800 images and 200 images, respectively.

B. Implementation

1) Evaluation metrics: precision: the fraction of retrieved figures that are relevant to the query:

$$precision = \frac{|\{relevant \quad images\} \bigcap \{retrieved \quad images\}|}{|\{retrieved \quad images\}|} \tag{4}$$

Recall: the fraction of the relevant documents that are successfully retrieved:

$$recall = \frac{|\{relevant \quad images\} \bigcap \{retrieved \quad images\}|}{|\{retrieved \quad images\}|} \tag{5}$$

F_1 measure: A measure that combines precision and recall is the harmonic mean of precision and recall:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6}$$

2) Hyper-parameters: The maximum number of weak classifier the model can use is 10.

3) Evaluation results: The experiments are shown in Table I.

TABLE I
Evaluation results of different parameters

depth	classifier number	precision	recall	f1-score	support
1	5	0.7051	0.7050	0.7050	200
1	10	0.7361	0.6150	0.5634	200
2	5	0.7703	0.7550	0.7523	200
2	10	0.8049	0.7800	0.7762	200
3	**5**	**0.8807**	**0.8800**	**0.8800**	**200**
3	10	0.7959	0.7950	0.7950	200
4	5	0.8451	0.8450	0.8450	200
4	10	0.8400	0.8400	0.8400	200
5	5	0.8551	0.8550	0.8550	200
5	10	0.8702	0.8700	0.8700	200

4) analysis: Our face detection system present good results. The result fluctuates with the changes of depth of the decision tree and number of weaker classifier, reaching the best with the trees depth of 3 and 5 weaker classifiers.

IV. Conclusion

This report shows the results of face classification based on AdaBoost algorithm. The results of our face detection system are promising but it is clear that many improvements are possible. And through this experiment, we found that some improvements could target the detection speed and the detection performances: The first improvement possible is certainly to modify the implementation of our scanning processing in order to take profit of the power of the cascade. The classification of a window needs only few processor instructions. Then the training of the cascade may be improved using more appropriated training sets. The problem is to find an optimal set with a large generalization power. The Bootstrapping principle seems to be particularly well chosen and it could be more used during the learning process to improve the robustness of the classifier.

REFERENCES

[1] Kgl B. The return of AdaBoost.MH: multi-class Hamming trees[J]. Computer Science, 2013.
[2] Joglekar, Sachin. "adaboost Sachin Joglekar's blog". codesachin.wordpress.com. Retrieved 3 August 2016.
[3] Meynet J, Popovici V, Thiran J. Fast Face Detection Using AdaBoost[J]. Epfl, 2003, 23(2):113119.