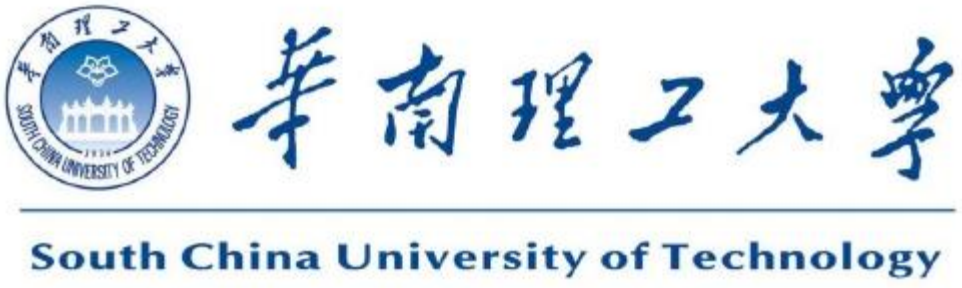

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Zilong Li ,Di Liu,Haijun You

Supervisor:
Qingyao Wu

Student ID：
201721045398, 201721045374,
201721045312

Grade:
Graduate

December 14, 2017

Face Classification Based on AdaBoost Algorithm

Abstract

We implemented a Face Classification algorithm Based On AdaBoost algorithm ,which use the Decision Tree Classifier as the base weak classifier .We use 670 image ,including 335 face images and 335 non face images as training images. And We achieved 92% accuracy in the validation dataset, which contains 165 face images and non face images.

I. INTRODUCTION

Face detection is a well studied problem in computer vision. It's main task is to distinguish the face from background, which contains two small tasks. The first small task is to locate the face proposal .And the second is to confirm whether this face proposal is a true face, also known as face classification.

In this paper we focus on the task two. We trained a face classifier based on AdaBoost algorithm ,which is a integrated learning classifier. The base classifier we used is decision tree classifier, which is very simple classifier only have 2 layers.

We use the provided dataset for training and validation .The whole dataset contains 1000 images ,including 500 face images as positive example ,500 nonface images as negative example. We separate the whole dataset for 2:1.So, the training images is 670 ,and the validation images is 330.

We use 100 weak decision tree classifier for boosting. The whole train process cost 251seconds. The final face classifier's accuracy is 88.49% in the validation dataset.

II. METHODS AND THEORY

2.1Adaboost

AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire, who won the 2003 Gödel 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.

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.

2.1.1 AdaBoost whole procedure

Given: $(x_1, y_1), \ldots, (x_1, y_1)$where$x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = \frac{1}{m}$

For $\mathrm{t} = 1, \ldots, \mathrm{T}$:

- Train weak learner using distribution D_t
- Get weak hypothesis h_t:$X \rightarrow$ {-1,+1}with error

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

- Choose $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$
- Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & if\ h_t(x_i) = y_i \\ e^{\alpha_t} & if\ h_t(x_i) \neq y_i \end{cases} = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Where $Z_t$is a normalization factor(chosen so that D_t will be a distribution).

$$Z_t = \sum_{i=1}^{N} D_t(i)\exp(-\alpha_t y_i h_t(x_i))$$

Output the final hypothesis:

$$\mathrm{H(x)} = \mathrm{sign}(\sum_{t=1}^{T} \alpha_t h_t(x))$$

2.2 Decision Tree

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making). This page deals with decision trees in data mining.

2.2.1 Decision tree constructing

Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items.[3] Different algorithms use different metrics for measuring "best". These generally measure the homogeneity of the target variable within the subsets. Some examples are given below. These metrics are applied to each candidate subset, and the resulting values are combined (e.g., averaged) to provide a measure of the quality of the split.

Gini impurity

Used by the CART (classification and regression tree) algorithm, Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. Gini impurity can be computed by summing the probability p_i of an item with label i being chosen times the probability $\sum_{k\neq i} 1-p_i$ of a mistake in categorizing that item. It reaches its minimum (zero) when all cases in the node fall into a single target category.

To compute Gini impurity for a set of items with J classes, suppose $i \in \{1,2,\dots,J\}$, and let p_i be the fraction of items labeled with class i in the set.

$$I_G(p) = \sum_{i=1}^{J} p_i \sum_{k\neq i} p_k = \sum_{i=1}^{J} p_i\,(1-p_i) = 1 - \sum_{i=1}^{J} p_i^2$$

Information gain

Used by the ID3, C4.5 and C5.0 tree-generation algorithms. Information gain is based on the concept of entropy from information theory.

Entropy is defined as below

$$H(T) = I_E(p_1, p_2, \dots, p_J) = -\sum_{i=1}^{J} p_i log_2 p_i$$

where $p_1, p_2,$ ...are fractions that add up to 1 and represent the percentage of each class present in the child node that results from a split in the tree.[4]

Variance reduction

Introduced in CART,[5] variance reduction is often employed in cases where the target variable is continuous (regression tree), meaning that use of many other metrics would first require discretization before being applied. The variance reduction of a node N is defined as the total reduction of the variance of the target variable x due to the split at this node:

$$\boldsymbol{I_V(N)} = \frac{\mathbf{1}}{|\boldsymbol{S}|^2}\sum_{i\in S}\sum_{j\in S}\frac{\mathbf{1}}{\mathbf{2}}(x_i - x_j)^2 - \left(\frac{\mathbf{1}}{|\boldsymbol{S_t}|^2}\sum_{i\in S_t}\sum_{j\in S_t}\frac{\mathbf{1}}{\mathbf{2}}(x_i - x_j)^2 + \frac{\mathbf{1}}{|\boldsymbol{S_f}|^2}\sum_{i\in S_f}\sum_{j\in S_f}\frac{\mathbf{1}}{\mathbf{2}}(x_i - x_j)^2\right)$$

Where S, S_t and S_f are the set of presplit sample indices, set of sample indices for which the split test is true, and set of sample indices for which the split test is false, respectively. Each of the above summands are indeed variance estimates, though, written in a form without directly referring to the mean.

III. EXPERIMENT

The dataset we use in training the face classification algorithm is provided by Yaofo Chen, which contains 500 face images and 500 nonface images .Some examples are as follow:

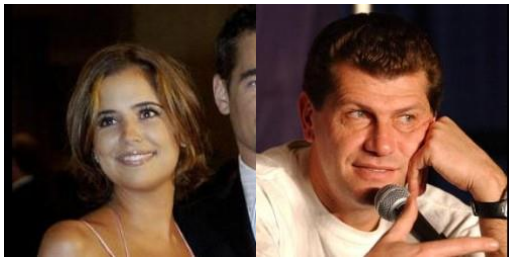

Figure 1:face image examples

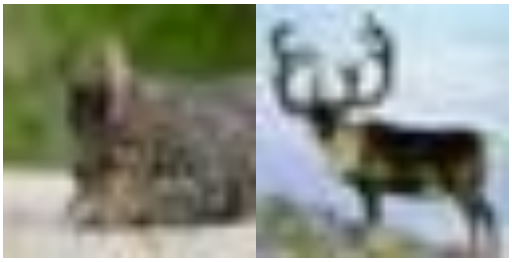

Figure 2:nonface image examples

We separate the whole dataset to 670 images and 330 nonface images. The ratio of positive examples to negative examples is 1:1.

We select the Decision Tree Classifier as the basic weak classifier. For a single Decision Tree Classifier ,it's splitter is "random",it's max depth is 1 .

The whole AdaBoost face classifier contains 100 weak classifier, cost 251seconds for training.

The Accuracy that the face classifier achieved in validation dataset is 88.49%, detail outcome is as follow:

	precision	recall	f1-score	Support
-1	0.92	0.87	0.89	179
1	0.85	0.91	0.88	151
avg / total	0.89	0.88	0.89	330

Table 1:face classifier report in validation dataset

IV. CONCLUSION

We propose a face classification algorithm based on AdaBoost algorithm .The AdaBoost algorithm is a Integrated learning algorithm .We use the simple decision tree classifier as the basic weak classifier. And we achieved not so bad result in the validation dataset. However ,how many weak classifier should a AdaBoost algorithm have and which type should the weak classifier be is still need to be discussed.

V. REFERENCES

[1].Freund, Yoav; Schapire, Robert E. A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. 1995. CiteSeerX: 10.1.1.56.9855.

[2].O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, 2nd Edition, Wiley, 2000, ISBN 978-0-471-05669-0

[3]Rokach, L.; Maimon, O. (2005). "Top-down induction of decision trees classifiers-a survey". IEEE Transactions on Systems, Man, and Cybernetics, Part C. 35 (4): 476–487. doi:10.1109/TSMCC.2004.843247.

[4] Witten, Ian; Frank, Eibe; Hall, Mark (2011). Data Mining. Burlington, MA: Morgan Kaufmann. pp. 102–103. ISBN 978-0-12-374856-0.
[5]Hastie, T., Tibshirani, R., Friedman, J. H. (2001). The elements of statistical learning : Data mining, inference, and prediction. New York: Springer Verlag.