IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 1 (Jan. - Feb. 2013), PP 35-37
www.iosrjournals.org

Ensemble Learning For Increasing Accuracy Data Models

Deepkanchan Nanasaheb Sonawane[1], Kavita Jain[2]
University Department of Computer Science, Mumbai, India

Abstract: *The volume of data has been growing tremendously in the past few years and is causing serious problems. The one of the major challenge is how to classify the newly emerging data and what basis.*
Finding effective methods for developing an ensemble of models has been an active research area of large-scale data mining in recent years. Models learned from data are often subject to some degree of uncertainty, for a variety of reasons. In classification, ensembles of models provide a useful means of averaging out error introduced by individual classifiers, hence increasing the accuracy of classification.

Keywords: *Ensemble Learning, Machine Learning, Naïve Bayes Classification, Bagging, data mining, weighted majority vote.*

I. INTRODUCTION

Data mining is about knowledge or model extraction from raw data. It can potentially lead to a model of the underlying data that allows us to make non-trivial predictions on new data. Models can often be treated as high-level abstractions of raw data obtained by the data mining process (i.e. learning algorithm) with a mining objective. Those modelsare often subject to various degrees of uncertainty. Depending on the mining objective, an infinite number of models could potentially result from a data mining task.
Ensemble learning refers to a collection of methods that learn a target function by training a number of individual learners and combining their predictions.

Method used for training ensembler:
Subsampling the training Samples:

Multiple hypotheses are generated by training individual classifiers on different datasets obtained by resampling a common training set

Structure of Ensemble Classifiers:
Parallel Structure:

All the individual classifiers are invoked independently, and their results are fused with a combination rule (e.g., average, weighted voting) or a metaclassifier.

II. METHODOLOGY

A. Structure of data:

We used 11 UCI datasets [2] for our experiments. We give their names, sizes,number of training instances used, number of test instances used and numbers of attributes and classes in Table 3.

B. Data Screening:

We divided each dataset roughly into four parts where three-fourth of the datasets is used as training samples to produce random boostraps and the remaining one-fourth of the datasets is used for testing.

C. Data Conversion:

The text (.csv) file obtained is converted to the data in the access database format. Further operations are performed on this dataset file.

A. NAÏVE BAYES ALGORITHM

a) Naïve Bayes Algorithm calculation

- Let **X** be a data sample ("*evidence*"), class label is unknown
- Let H be a *hypothesis* that X belongs to class C
- Classification is to determine P(H|**X**), the probability that the hypothesis holds given the observed data sample **X**
- P(H) (*prior probability*), the initial probability
- P(**X**): probability that sample data is observed
- P(**X**|H) (*posteriori probability*), the probability of observing the sample **X**, given that the hypothesis holds

- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes theorem
- Informally, this can be written as posteriori = likelihood x prior/evidence
- Predicts **X** belongs to C_2 iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the *k* classes

b) Naïve Bayes Algorithm description

Generate classifier given of training set

Input:

- Samples: Sample training set, expressed by discrete attribute values.
- attribute_list: Candidate attributes list.

Output:

A Classifier rule.

Steps: (By pseudo-code)

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector $\mathbf{X} = (x_1, x_2, \dots, x_n)$
- Suppose there are *m* classes $C_1, C_2, \dots, C_m$.
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem
- Since P(X) is constant for all classes, only
- needs to be maximized
- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):
- This greatly reduces the computation cost: Only counts the class distribution

B. BAGGING ALGORITHM

Input:

- Training data S with correct labels $W=\{w_1,\dots..,w_c\}$ representing C classes.
- Weak Learning Algorithm, WeakLearn,
- Integer T specifying number of iterations.
- Percent (or fraction) F to create bootstrapped training data
- Do t=1,…T

1. Take abootstrapped replica S_t by randomly drawing F percent of S.
2. Call weaklearn with S_t and receive the hypothesis(classifier)h_t.
3. Add h_t to ensemble E. End

C. VOTING METHOD:

Voting is a well-known aggregation procedure for combining opinions of voters in order to determine a consensus, an agreement on a given issue, within a given time frame.

Test:

Simple Majority Voting:

Given unlabeled instance x

1. Evaluate the ensemble $E=\{h_1\dots..h_T\}$ on x.
2. Let $v_{t,j}=1$, if ht picks class w_j
   0, otherwise

(be the vote given to class w_j by classifier ht.)

3. Obtain the vote received by each class,
   $$V_j=\sum^{T}_{t=1} V_{t,j} \quad j=1,\dots.,C.$$
4. Choose the class that receives the highest total Vote as the final classification.

Weighted Majority Voting:

For weighted voting system, we consider the classifiers $(D_1; D_2; : : : ; D_n)$ with accuracies $(p_1; p_2; : : : ; p_n)$; respectively. Then, let $d_{i,j}$ be defined in the following way:

$d_{i,j} = 1$, if the classifier D_i labels x in the class, and

$d_{i,j} = 0$, otherwise.

In case of weighted voting, the discriminant function for class is given as:
$g_j (x) = \sum^n_{i=1} b_i d_{i,j}$
Where the weight b_i corresponds to the classifer D_i

1. Initialize all bootstraps to weight 1.
2. for each round,

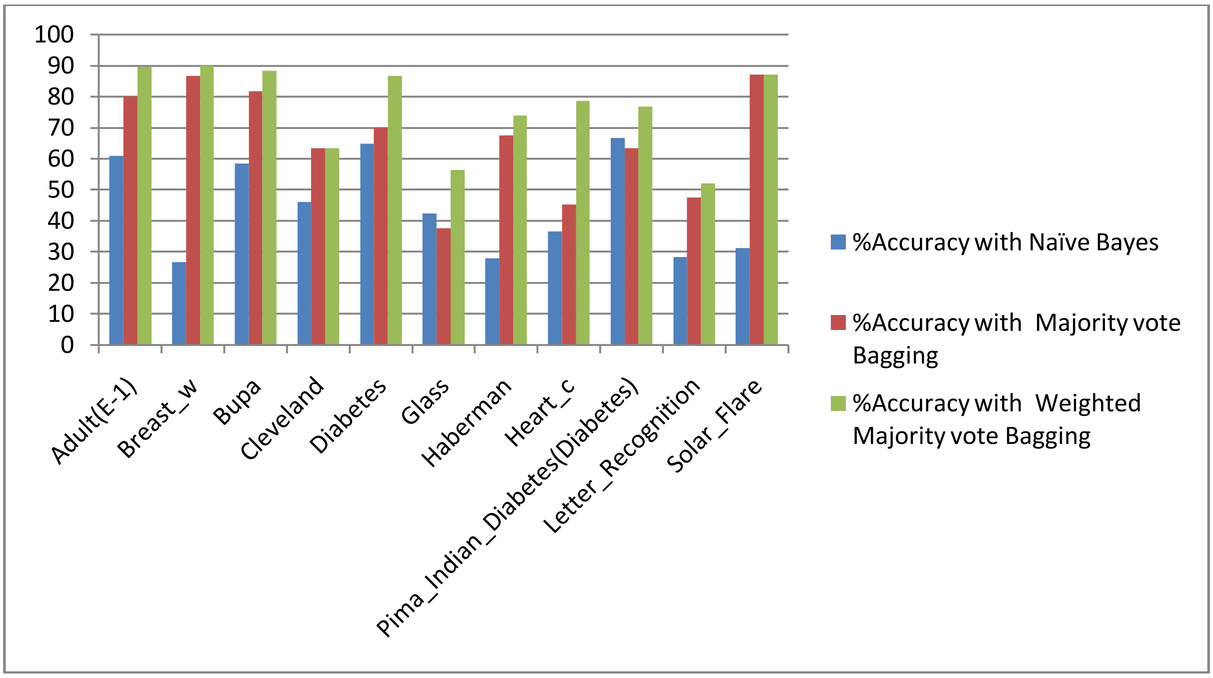
A) poll all the experts and predict based on a weighted majority vote of their predictions.
B) cut in half the weights of all experts that make a mistake. (no rewards.)
In a weighted majority voting system, the class label is chosen for x if $g_k(x) = \max_{j=1;:::;n} g_j (x) = \sum^n_{i=1} b_i d_{i,k}$:

III. CLASSIFICATION AND ANALYSIS

From this it is possible to see that classification with only Naïve Bias produced poor results. After applying bagging the considerable increase in actual classification is seen.
Bagging with majority vote increases the accuracy to considerable amount.

But the accuracy can still be increased if applied bagging with weighted majority vote as in the proposed algorithm.

The method requires that data be discrete.



REFERENCES

[1] www.abeautiful.com
[2] www.ics.**uci**.edu/~mlearn/ C. Blake and C. Merz. UCI repository of machine
[3] www.ensembleLearning\articles
[4] http://www.cs.ucla.edu.
[5] http://itee.uq.edu.au
[6] Margaret H. Dunham, S. Sridhar Data Mining:Introductory and AdvancedTopics, Pearson Education.
[7] Lectures on Machine Learning by www.coursera.org.