Data Mining and Warehousing - Assignment 5

Indian Institute of Information Technology, Allahabad

IIT2018142 IIT2018146 IT2018150 IIT2018184 IIT2018185

Abstract: Support Vector Machine is a supervised machine learning algorithm mainly used to solve classification problems. It finds the optimal hyperplane that best classifies the dataset into 2 or more classes. SVM boosting algorithms are used to get better results. We implement the idea of Markov resampling for Boosting methods described in a paper.

I. INTRODUCTION

Boosting is to obtain base learners by adjusting the weights of training examples. SVM Boosting algorithm which we implement in this paper have better accuracy , smaller misclassification rates, less total time of sampling and training compared to three classical AdaBoost algorithms: Gentle AdaBoost, Real AdaBoost, Modest AdaBoost. The main idea of Markov resampling proposed in this paper is to generate uniformly ergodic Markov chain multiple times.

We apply Boosting algorithm based on Markov resampling to Support Vector Machine (SVM), and introduce two new resampling based Boosting algorithm: **Improvised SVM-Boosting based on Markov resampling (ISVM-BM)**. Compared with SVM-BM, ISVM-BM uses the support vectors to calculate the weights of base classifiers. SVM Boosting algorithm have better accuracy, smaller misclassification rates, less total time of sampling and training compared to three AdaBoost algorithms: Gentle AdaBoost, Real AdaBoost, Modest AdaBoost. In code we trained our model on basis of algorithm and print metrics like accuracy, misclassification rate, f1-score, recall etc. to analyze the result.

II. DATASET DESCRIPTION

We have used 1 dataset for training and testing purpose. Dataset was provided by UCI machine learning repository. The first dataset consists of 20000 instances of 26 Capital letters in the english alphabet. The images are based on 20 different fonts where each letter within these 20 fonts is randomly distorted to produce 20,000 unique samples. We train our SVM on the 80% part of dataset i.e. first 16000 samples and use the rest 20% (4000 samples) for testing.

III. ALGORITHM

Algorithm 2: ISVM-BM

Input: D_{train}, n_2, q, N, T

Output: $\mathrm{sign}(f_T) = \mathrm{sign}(\sum_{t=1}^{T} \hat{\alpha}_t g_t)$

Draw randomly samples $D_0 = \{z_i\}_{i=1}^{N}$ from D_{train}, train D_0 by algorithm (8) and obtain a classification function g_0, draw randomly a example z from D_{train} and $z_1 \leftarrow z$, let $t \leftarrow 1$

while $t \le T$ **do**

  $i \leftarrow 1$, $n_1 \leftarrow 0$

  while $i \le$ N **do**

    Draw randomly a sample z_* from D_{train},

    $p_t^{i+1} \leftarrow \min\{1, e^{-\ell(g_{t-1}, z_*)}/e^{-\ell(g_{t-1}, z_i)}\}$

    if $n_1 > n_2$ **then**

      $p_t^{i+1} \leftarrow \min\{1, qp_t^{i+1}\}$, $z_i \leftarrow z_*$, $D_t \leftarrow z_i$, $i \leftarrow i+1$, $n_1 \leftarrow 0$

    end

    if $p_t^{i+1} \equiv 1$ and $y_* y_i = 1$ **then**

      $p_t^{i+1} \leftarrow e^{-y_* g_{t-1}}/e^{-y_i g_{t-1}}$

    end

    if rand(1) $< p_t^{i+1}$ **then**

      $z_i \leftarrow z_*$, $D_t \leftarrow z_i$, $i \leftarrow i+1$, $n_1 \leftarrow 0$

    end

    if z_* is not accepted **then**

      $n_1 \leftarrow n_1 + 1$

    end

  end

  Obtain Markov chain $D_t = \{z_i\}_{i=1}^{N}$. Train D_t by algorithm (8) and obtain another classification function g_t. Denote support vectors as D_{SV}^t.

  $e'_t \leftarrow \mathrm{P}\big(Y \neq \mathrm{sign}(g_t(X)) | \cup_{j=1}^{t} D_{SV}^{j}\big)$, $\hat{\alpha}_t \leftarrow (1/2) * \log\big((1-e'_t)/e'_t\big)$,

  $z_1 \leftarrow z_*$, $t \leftarrow t+1$

  if $\hat{\alpha}_t < 0$ **then**

    $t \leftarrow t-1$

  end

end

III. RESULT

We return the model after training part. Then we classify our testing input. For each letter in 26 alphabets, we print the parameters like accuracy, misclassification rate, f1-score, recall etc. and analyze the result.

Misclassification Rates						
Kernel	KPCA	SVDD	OCSVM	OCSSVM	OCSSVM with SMO	ISVM-BM
Linear	0.02	0.09	0.01	0.07	0.04	0.502
RBF	0.05	0.07	0.14	0.09	0.04	0.769
Intersection	0.18	0.01	0.04	0.26	0.22	
Hellinger	0.01	0.02	0..02	0.13	0.1	
Sigmoid						0.950
Polynomial						0.564