Data Mining

Group Members:

1. Prabal(IIT2018140)
2. Nikhil Kumar(IIT2018152)
3. Sagar Kumar(IIT2018154)
4. Kartik Nema(IIT2018156)
5. Bhupendra(IIT2018163)
6. Prakhar Srivastava(IIT2018172)

SVM-Boosting based on Markov resampling

Introduction: We know the performance of SVM. In some cases it takes a lot of time. So this paper implements SVM with Markov sampling and other techniques. This paper also compares the performance among them.While Boosting is to obtain base learners by adjusting the weights of training examples. The most famous Boosting method is AdaBoost (Adaptive Boosting), which was introduced by Freund and Schapire in Freund and Schapire (1996, 1997). Different from Bagging, the examples misclassified by the last base learner will receive more attention in the next train, and repeat the process above, until up to the given number of iterations. Breiman proved that AdaBoost algorithm based on decision (complete) tree can converge to the Bayes risk as the size of training examples is sufficiently big. Jiang not only presented the examples that AdaBoost has prediction error asymptotically suboptimal as the number of iterations is enough big, but also pointed that some regularization methods may make the prediction error close to the Bayes risk when the size of training examples.

Keywords: SVM,Markov Sampling,ISVM-BM,SVM-BM

Algorithm:

Algorithm 1: SVM-BM

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

Output: $\text{sign}(f_T) = \text{sign}(\sum_{t=1}^{T} \alpha_t g_t)$

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 sample z from D_{train}.

$z_1 \leftarrow z$, let $t \leftarrow 1$

while $t \leq T$ **do**
- $i \leftarrow 1$, $n_1 \leftarrow 0$
- **while** $i \leq$ 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.
- $e_t \leftarrow P(Y \neq \text{sign}(g_t(X)) | D_{train})$.
- $\alpha_t \leftarrow (1/2) * \log((1 - e_t)/e_t)$.
- $z_1 \leftarrow z_*$, $t \leftarrow t+1$
- **if** $\alpha_t < 0$ **then**
  - $t \leftarrow t-1$
- **end**

end

Algorithm 2: ISVM-BM

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

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

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 $t \leq T$ **do**

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

  while $i \leq$ 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 $p_t^{i+1} = 1$ 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.

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

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

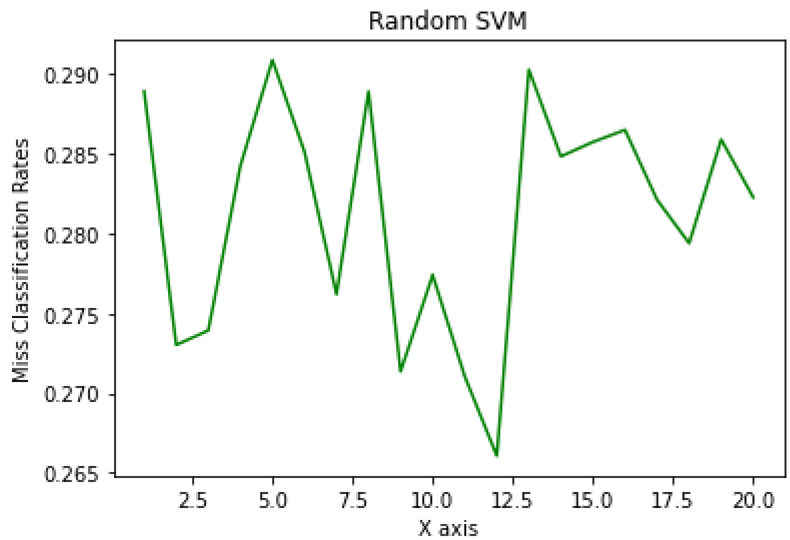
  if $\hat{\alpha}_t < 0$ **then**

    $t \leftarrow t - 1$

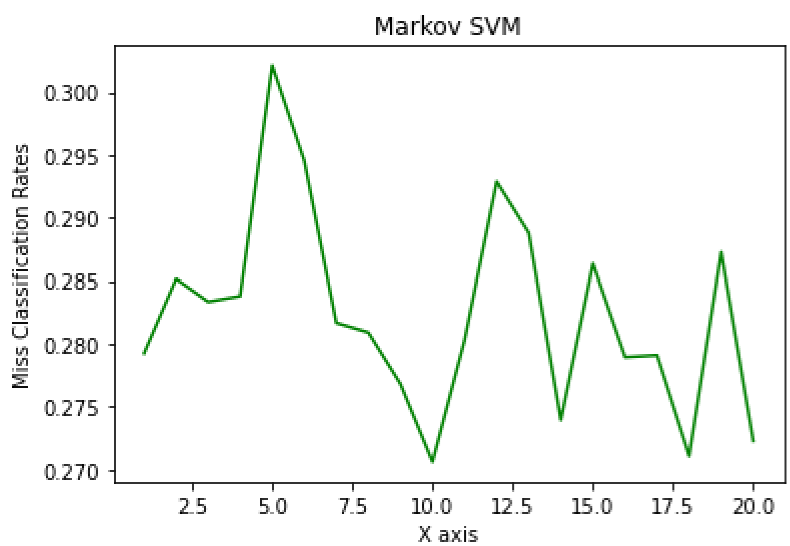
  end

end

Results:



Mean misclassification (random sample) for letter (28.121969696969696, 0.7067247981443773)



Mean misclassification rate (markov sample) for letter (28.245454545454546, 0.7905200482847531)

Conclusion: In this paper, we introduced the idea of resampling for Boosting algorithms. We firstly proved that the resampling-based Boosting algorithm with general convex loss function is consistent and established the fast learning rate for resampling-based Boosting algorithm. To our knowledge, these results are the first results on this topic. We also applied the Boosting algorithm based on resampling to the classical classification algorithm, SVM, and proposed the SVM Boosting based on Markov resampling algorithm (SVM-BM).