K-Times Markov Sampling for SVMC

Rahul ReddyMuppidi-IIT2018103

Lokesh Naik- IIT2018104

A Prathyush-IIT2018124

Mallampalli Maheswarnath-IIT2018137

Gowni Sai Pavan-IIT2018139

B Tech(IT)-6th sem,IIIT Allahabad

I. Introduction

Many machine learning applications, such as market prediction, system diagnosis, and speech recognition, are inherently temporal in nature, and consequently not independent and identically distributed. processes . Therefore, the relaxations of such . assumptions for SVMC have to be considered. They studied the generalization ability of SVMC with uniformly ergodic Markov chain samples, but the obtained learning rate is not optimal. The optimal learning rate of Gaussian kernels SVMC with samples by using the strongly mixing property of the Markov chain. The optimal learning rate of SVMC with u.e.M.c. samples and presented the numerical studies on the performance of SVMC with Markov sampling. Although the SVMC with Markov sampling introduced has smaller misclassification rates,its total time of sampling and training is longer compared with the classical SVMC based on randomly independent sampling. The main purpose of using k-times markov sampling SMVC is that we need to reduce the sampling and training time of SVMC with Markov sampling, at the same time keeping its smaller misclassification rates.

Markov sampling has three advantages at the same time compared with the classical SVMC and the SVMC with Markov sampling:

(1) The misclassification rates are smaller.

(2) The total time of sampling and training is less.

(3) the obtained classifiers are more sparse.

To have a better showing the performance of SVMC with k-times Markov sampling, we also give some discussions for the cases of unbalanced training samples and large-scale training samples.

II. Algorithm

Input: ST , N, k, q, n2

Output: sign(fk)

1) Draw randomly N samples Siid := {zj}N j=1 from ST . Train Siid by SVMC and obtain a eliminary
learning model f0. Let i = 0.

2) Let Ni = 0, t = 1.

3) Draw randomly a sample zt from ST , called it the current sample. Let Ni = Ni + 1.

4) Draw randomly another sample z∗ from ST , called it the candidate sample. Calculate the ratio α, α =
e−(fi,z∗) /e−(fi,zt)

5) If α = 1, yt y∗ = 1 accept z∗ with probability α1 = e−y∗ fi /e−yt fi . If α = 1 and yt y∗ = −1 or α < 1, accept z∗ with probability α. If there are n2 candidate samples can not be accepted continually, then set α2 = qα and accept z∗ with probability α2. If z∗ is not accepted, go to

Step 4, else let $z_{t+1} = z*$, $N_i = N_i + 1$ (if α (or α_1,α_2) is greater than 1, accept $z*$ with probability 1).
6) If $N_i < N$, return to Step 4, else we obtain N Markov chain samples SMar. Let $i = i + 1$.

Train SMar by SVMC and obtain a learning model f_i .
7) If $i < k$, go to Step 2, else output sign(f_k)

III. Result

Kernel	KPCA	SVDD	OCSVM	OCSSVM	OCSSVM with SMO	KT_SVM
Linear	0.02	0.09	0.01	0.07	0.04	0.79
RBF	0.05	0.07	0.14	0.09	0.04	0.89
Intersection	0.18	0.01	0.04	0.26	0.22	
X^2	0.18	0.0	0.02	0.18	0.17	0.84

IV. Conclusion:

As we can see in the result, there's improvement in the accuracy while using this approach.