**INDIAN INSTITUTE OF INFORMATION**

**TECHNOLOGY, ALLAHABAD**

**B. Tech, VI Semester**

**Report - Group Assignment 1**

**Data Mining and Warehousing**

**Group Members:**

1. **IIT2018155 (ABHINAV BANSAL)**
2. **IIT2018168 (NEELABH GUPTA)**
3. **IIT2018174 (ANKIT RAJ)**
4. **IIT2018175(ASHISH PATEL)**
5. **IIT2018177(SHUBHAM SONI)**

**Title:** k-Times Markov Sampling for SVMC

**Author:** Bin Zou, Chen Xu, Yang Lu, Yuan Yan Tang, Fellow, IEEE, Jie Xu, and Xinge You

**Introduction-**

Support vector machines (SVM) is one of the most widely used learning algorithms for classification problems. Although SVM has good performance in practical applications, it has high algorithmic complexity as the size of training samples is large. SVMC with Markov sampling has smaller misclassifications rates, the total time of sampling and training is longer compared to the classical SMC based on randomly independent sampling. To improve the learning performance of the classical SVMC, we use the SVMC algorithm based in k-times Makov sampling and present the numerical studies on the learning performance of SVMC with k-times Markov Sampling.

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

* The misclassification rates are smaller;
* The total time of sampling and training is less;
* The obtained classifiers are more sparse.

**IMPORTANT TERMINOLOGY:**

**SVM:** The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. It is a binary classification algorithm that transforms data and finds the best boundary between the possible outputs depending on the transformations. SVM is used because

* It works really well with a clear margin of separation
* It is effective in high dimensional spaces.
* It is effective in cases where the number of dimensions is greater than the number of samples.
* It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

**ALGORITHM:-**

**Input:** ST , N, k, q, n2

**Output**: sign( fk )

1: Initially we draw randomly N samples Siid := {zj}N j=1 from ST , then after training Siid by SVMC and obtain a preliminary learning model f0. Let i = 0.

2: Assuming ( N+) = 0, (N− ) = 0, t = 1.

3: Draw randomly a sample zt from ST , called it the current sample.

Let increment (N+) and (N-) according to the label zt,if the label of zt is +1, then increment (N+) or if the label of zt is −1,increment (N-).

4: Draw randomly another sample z∗ from ST , called it the candidate sample, and 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 zt+1 = z∗, N+ = N+ + 1 if the label of zt+1 is +1 and N+ < N/2, or let zt+1 = z∗, N− = N−+1 if the label of zt+1 is −1 and N− < N/2 (if the value α (or α1, α2) is bigger than 1, accept the candidate sample z∗ with probability 1).

6: If N+ +N− < 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 fi .

7: If i < k, go to Step 2, else output sign( fk ).

**RESULT:-**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Kernel | KPCA | SVDD | OCSVM | OCSSVM | OCSSVM  with smo | k\_MS\_SVM |
| Linear | 0.02 | 0.09 | 0.01 | 0.07 | 0.04 | 0.855 |
| RBF | 0.05 | 0.07 | 0.14 | 0.09 | 0.04 | 0.94225 |
| Hellinger | 0.01 | 0.02 | 0.02 | 0.13 | 0.10 | 0.8332 |
| chi\_square | 0.18 | 0.0 | 0.02 | 0.18 | 0.17 | 0.891 |