INDIAN INSTITUTE OF INFORMATION TECHNOLOGY ALLAHABAD

Subject – Data Mining and Warehousing

Topic – k-Times Markov Sampling for SVMC

Report By -

IIT2018108 – Ravi Kumar Sharma

Assignment :

You have to understand the algorithm proposed in the paper "k -Times Markov Sampling for SVMC ".
Run the algorithm on the shared given two datasets and show the accuracy in terms of the attached image table: (make one more column in the last name KT_SVM with the new algorithm and give the result).

Try to run the algorithm using different kernels as attached image table.

1) http://host.robots.ox.ac.uk/pascal/VOC/voc2012/ - pascal

2) https://archive.ics.uci.edu/ml/datasets/Letter+Recognition – letter

Kernel	KPCA	SVDD	OCSVM	OCSSVM	OCSSVM with SMO
Linear	0.02	0.09	0.01	0.07	0.04
RBF	0.05	0.07	0.14	0.09	0.04
Intersection	0.18	0.01	0.04	0.26	0.22
Hellinger	0.01	0.02	0.02	0.13	0.10
χ^2	0.18	0.0	0.02	0.18	0.17

Introduction :

support vector machine classification (SVMC) algorithm are usually based on the assumption of independent and identically distributed (i.i.d.) samples.

Instruction to run the code

step #1. open the code in google colab or jupyter notebook.
step #2. upload the required dataSet uploaded with this code.
step #3. run the all the codes of whole block serialWise.
step #4. Look at the result or output of the code of all sections.

Algorithm 1 :

Algorithm 1 SVMC Algorithm Based on k Times Markov Sampling for Balanced Training Samples

Input: S_T, N, k, q, n_2
Output: $sign(f_k)$

1: Draw randomly N samples $S_{iid} := \{z_j\}_{j=1}^N$ from S_T. Train S_{iid} by SVMC and obtain a preliminary learning model f_0. Let $i = 0$.
2: Let $N_+ = 0$, $N_- = 0$, $t = 1$.
3: Draw randomly a sample z_t from S_T, called it the current sample. Let $N_+ = N_+ + 1$ if the label of z_t is $+1$, or let $N_- = N_- + 1$ if the label of z_t is -1.
4: Draw randomly another sample z_* from S_T, called it the candidate sample, and calculate the ratio α, $\alpha = e^{-\ell(f_i, z_*)}/e^{-\ell(f_i, z_t)}$.
5: If $\alpha \geq 1$, $y_t y_* = 1$ accept z_* with probability $\alpha_1 = e^{-y_* f_i}/e^{-y_t f_i}$. If $\alpha = 1$ and $y_t y_* = -1$ or $\alpha < 1$, accept z_* with probability α. If there are n_2 candidate samples can not be accepted continually, then set $\alpha_2 = q\alpha$ and accept z_* with probability α_2. If z_* is not accepted, go to Step 4, else let $z_{t+1} = z_*$, $N_+ = N_+ + 1$ if the label of z_{t+1} is $+1$ and $N_+ < N/2$, or let $z_{t+1} = z_*$, $N_- = N_- + 1$ if the label of z_{t+1} is -1 and $N_- < N/2$ (if the value α (or α_1, α_2) 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 S_{Mar}. Let $i = i + 1$. Train S_{Mar} by SVMC and obtain a learning model f_i.
7: If $i < k$, go to Step 2, else output $sign(f_k)$.

Algorithm 2 :

Algorithm 2 SVMC Algorithm Based on k Times Markov Sampling for Unbalanced Training Samples

Input: S_T, N, k, q, n_2
Output: $sign(f_k)$

1: Draw randomly N samples $S_{iid} := \{z_j\}_{j=1}^N$ from S_T. Train S_{iid} by SVMC and obtain a preliminary learning model f_0. Let $i = 0$.
2: Let $N_i = 0$, $t = 1$.
3: Draw randomly a sample z_t from S_T, called it the current sample. Let $N_i = N_i + 1$.
4: Draw randomly another sample z_* from S_T, called it the candidate sample. Calculate the ratio α, $\alpha = e^{-\ell(f_i, z_*)}/e^{-\ell(f_i, z_t)}$.
5: If $\alpha = 1$, $y_t y_* = 1$ accept z_* with probability $\alpha_1 = e^{-y_* f_i}/e^{-y_t f_i}$. If $\alpha = 1$ and $y_t y_* = -1$ or $\alpha < 1$, accept z_* with probability α. If there are n_2 candidate samples can not be accepted continually, then set $\alpha_2 = q\alpha$ and accept z_* with probability α_2. 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 S_{Mar}. Let $i = i + 1$. Train S_{Mar} by SVMC and obtain a learning model f_i.
7: If $i < k$, go to Step 2, else output $sign(f_k)$.

Results :

1. letter-recognition dataset

Kernel	KPCA	SVDD	OCSVM	OCSSVM	OCSSVM with SMO	KT_SVM
Linear	0.02	0.09	0.01	0.07	0.04	0.8121
RBF	0.05	0.07	0.14	0.09	0.04	0.869
Intersection	0.18	0.01	0.04	0.26	0.22	0.0192
Hellinger	0.01	0.02	0.02	0.13	0.10	0.7092
chi_square	0.18	0.0	0.02	0.18	0.17	0.8495

2. Pascal dataset

Kernel	KPCA	SVDD	OCSVM	OCSSVM	OCSSVM with SMO	KT_SVM
Linear	0.02	0.09	0.01	0.07	0.04	0.2160
RBF	0.05	0.07	0.14	0.09	0.04	0.3092
Intersection	0.18	0.01	0.04	0.26	0.22	TOO MUCH TIME
Hellinger	0.01	0.02	0.02	0.13	0.10	0.18
chi_square	0.18	0.0	0.02	0.18	0.17	0.2336