

**Email:** hts-users@sp.nitech.ac.jp

**Title:** Bug Report on ApplyWindow Function in HMGenS

HTS working group,

Thank you for your excellent work on HTS.

I find a moderate bug in ApplyWindow() in HMGenS V2.2 and V2.3-alpha. This bug is about how to calculate dynamics at time  $t=1, T$ , where  $t$  is time index, and  $T$  is total frame number. The current implementation of ApplyWindow() virtually truncated any involved frame when its index is out of the valid range,  $\{1, 2, \dots, T\}$ , but such treatment is not proper. However, the same problem does not occur in window.pl in HTS-demo release, as window.pl handles this situation properly.

### Analysis

A concrete example will make it clear. The default configuration in HTS-demo release for calculating dynamics is

$$\begin{cases} \Delta^{(0)}\mathbf{c}_t = \mathbf{c}_t \\ \Delta^{(1)}\mathbf{c}_t = -\frac{1}{2}\mathbf{c}_{t-1} + \frac{1}{2}\mathbf{c}_{t+1} \\ \Delta^{(2)}\mathbf{c}_t = \mathbf{c}_{t-1} - 2\mathbf{c}_t + \mathbf{c}_{t+1} \end{cases} \quad (1)$$

where  $t=1, 2, \dots, T$ . Those are the zero-th to the second order central differences.

When calculating  $\Delta^{(n)}\mathbf{c}_t, n=1, 2, t=1, T$ , frame indexes  $t=0, T+1$  are involved but do not valid, thus some heuristic or a variant of Eq. (1) must be used instead. We simply take  $\Delta^{(1)}\mathbf{c}_1$  as an illustrative example here, and  $\Delta^{(2)}\mathbf{c}_1, \Delta^{(1)}\mathbf{c}_T, \Delta^{(2)}\mathbf{c}_T$  can be analyzed similarly.

Perl scripts window.pl in ./data/scripts in HTS-demo takes heuristics of “using the nearest, valid frame index instead”, or “time cut-off method”, i.e.,

$$\Delta^{(1)}\mathbf{c}_1 = -\frac{1}{2}\mathbf{c}_0 + \frac{1}{2}\mathbf{c}_2 \approx -\frac{1}{2}\mathbf{c}_1 + \frac{1}{2}\mathbf{c}_2. \quad (2)$$

C Function ApplyWindow() in HMGenS in HTS toolkit takes heuristics of “ignoring any invalid frame”, or “zero assumption method”, i.e.,

$$\Delta^{(1)}\mathbf{c}_1 = -\frac{1}{2}\mathbf{c}_0 + \frac{1}{2}\mathbf{c}_2 \approx \frac{1}{2}\mathbf{c}_2. \quad (3)$$

Obviously, Eq. (3) gives wrong dynamics. The simplest bug fix is to take the same heuristics used in window.pl in Eq. (2).

### Experiment

Essential condition: HMGenS -c 0, GV turned off.

The following “probByFrame”,  $b_{q_t}(\mathbf{o}_t)$ , is output in function OutProb() in HGen.c

### Before fixed:

The following probByFrame is incorrect when  $t=1, T$ .

Generating Label CASIA\_ZXY\_00001.lab

Total number of frames = 640

PdfStream[1]: 640 frames

PdfStream[2]: 247 frames

```
t = 1, probByFrame = -8.958799e+005
t = 2, probByFrame = 3.871157e+002
t = 3, probByFrame = 3.650192e+002
t = 4, probByFrame = 3.734155e+002
t = 5, probByFrame = 3.748257e+002
...
t = 636, probByFrame = 3.751290e+002
t = 637, probByFrame = 3.749175e+002
t = 638, probByFrame = 3.566011e+002
t = 639, probByFrame = 3.330691e+002
t = 640, probByFrame = -2.084838e+005
```

### After fixed:

The following probByFrame are corrected when  $t=1, T$ .

```
t = 1, probByFrame = 3.447705e+002
t = 2, probByFrame = 3.871157e+002
t = 3, probByFrame = 3.650192e+002
t = 4, probByFrame = 3.734155e+002
t = 5, probByFrame = 3.748257e+002
...
t = 636, probByFrame = 3.751290e+002
t = 637, probByFrame = 3.749175e+002
t = 638, probByFrame = 3.566011e+002
t = 639, probByFrame = 3.330691e+002
t = 640, probByFrame = 3.756874e+002
```

### Implementation

This report and my bug fix are supplied in attachment for reference.

### Discussion

Please let me know if I am not correct. Thank you very much!

Yang Wang

Email: yangwang@nlpr.ia.ac.cn