Details of Stationarity Measure ψ

November 12, 2020

We have developed a measure of audio signal stationarity called ψ. It is based on changes in 13 statistics: log power and the lag 1 to 12 normalized auto-covariance values. Pseudocode that calculates ψ is provided below, and text that further describes that code is provided here. Note that four separate blocks of pseudocode are provided.

The input audio vector $\mathbf{x}$ has sample rate $f_s = 48,000$ smp/s. We calculate each statistic over 5 ms (240 smp). This window length represents a compromise between robust statistics (which benefits from more samples) and temporal resolution (which benefits from fewer samples). The result is the 13 by $nSmp$ matrix $\mathbf{S}$ that contains the 13 statistics in each of the $nSmp$ columns. (Log power is the first statistic and the normalized auto-covariance values follow in order.) Note that the value of $nSmp$ is 251 $(239 + 12)$ less than than the length of input audio signal $\mathbf{x}$. We advance the 5 ms window one sample for each new calculation of statistics. The result is thirteen sequences of statistics.

The next steps (lines 5-18) find blocks of contiguous windows where none of the statistics changes by more than a specified value. These values are stored in the 13 by 1 vector **thresholds**. These values are 5 dB for log power and 0.5 for the normalized auto-covariance values. The blocks are found by extracting an initial set of statistics (line 6) and comparing these with a second set of statistics from a later time (lines 10 and 11) while continually increasing that later time until at least one of the 13 thresholds is met or exceeded (lines 9-13). The averaged log power (line 14) and length (line 15) of each of these blocks is recorded. The process (lines 5-18) produces a total of $nBlocks$ blocks.

These blocks are then tested (line 22) to remove low power blocks (irrelevant from an audio signal perspective) and blocks shorter than 10 ms (where the 5 ms windows must overlap, thus compromising tests for changes between windows). The lengths of the blocks that survive these tests are accumulated (line 23) and the number of blocks that survive is counted (line 24). These two values allow the computation of the average duration of the surviving blocks, L_b (line 27). This is a measure of the *average duration* of approximate stationarity (in samples). They also allow calculation of the fraction, R_b of the tested audio samples that are in a surviving block (line 28) and this is a unitless measure of the *level* of approximate stationarity. Finally we multiply these duration and level factors to arrive at an index of stationarity ψ that has units of time.

Algorithm 1 Stationarity Index ψ for audio signal $\mathbf{x}$

```
 1: $\mathbf{x} \leftarrow \mathbf{x}$ / std($\mathbf{x}$)
 2: $\mathbf{S} \leftarrow$ calcStats ($\mathbf{x}$)
 3: $start = 1$
 4: $j = 1$
 5: while $start < nSmp$ do
 6:     $\mathbf{refStats} = \mathbf{S}(:, start)$ (Extract column of statistics from $\mathbf{S}$)
 7:     $noChange = true$;
 8:     $i = 1$;
 9:     while $start + i \leq nSmp$ AND $noChange$ do
10:         $\mathbf{testStats} = \mathbf{S}(:, start + i)$ (Extract column of statistics from $\mathbf{S}$ )
11:         $noChange = \text{allWithinThreshold}(\mathbf{refStats}, \mathbf{testStats}, \mathbf{thresholds})$
12:         $i \leftarrow i + 1$
13:     end while
14:     $\mathbf{power}(j) = \text{mean}(\mathbf{S}(1, start : start + i - 1))$
15:     $\mathbf{length}(j) = (i - 1)$
16:     $j \leftarrow j + 1$
17:     $start \leftarrow start + i - 1$
18: end while
19: $lengthSum = 0$
20: $nValidBlocks = 0$
21: for j=1 to nBlocks do
22:     if $-30 < \mathbf{power}(j)$ AND $.01 < \mathbf{length}(j)/f_s$ then
23:         $lengthSum \leftarrow lengthSum + \mathbf{length}(j)$
24:         $nValidBlocks \leftarrow nValidBlocks + 1$
25:     end if
26: end for
27: $L_b = lengthSum/nValidBlocks$
28: $R_b = lengthSum/(nSmp - 1)$
29: $\psi = L_b R_b / f_s$
30: Return $\psi$
```

Algorithm 2 calcStats ($\mathbf{x}$)

```
1: for $i = 1, 2, \ldots$ do
2:     Calculate $10\log_{10}$(variance($x_i$ to $x_{i-1+240}$))
3:     for $j = 1, 2, \ldots 12$ do
4:         Calculate lag $j$ normalized auto-covariance($x_i$ to $x_{i-1+240+12}$))
5:     end for
6:     Store all 13 statistics in column $i$ of $\mathbf{S}$
7: end for
8: Return $\mathbf{S}$
```

Algorithm 3 normalizedAutoCovariance($\mathbf{x}, lag$)

```
1: $\overline{x}$ =mean($x_1$ to $x_{240+lag}$)
2: $\mathbf{x} \leftarrow \mathbf{x} - \overline{x}$
3: $\rho = \frac{\sum_{i=1}^{240} x_i x_{i+lag}}{\sqrt{\sum_{i=1}^{240} x_i^2 \sum_{i=1}^{240} x_{i+lag}^2}}$
4: Return $\rho$
```

Algorithm 4 allWithinThreshold ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{t}$)

```
1: allWithin $\leftarrow$ true
2: for $i = 1, 2, \ldots, 13$ do
3:     if $t_i \leq |x_i - y_i|$ then
4:         allWithin $\leftarrow$ false
5:     end if
6: end for
7: Return allWithin
```

```
function psi = stationarityIndex(x)
%x is a column vector of audio samples with fs=48k
%psi is a measure of stationarity, units are seconds
%Written October 2020 by S. Voran
%Institute for Telecommunication Sciences
%Boulder, Colorado, US
%svoran@ntia.gov
fs=48000;
thresholds = [5; .5*ones(12,1)];
x=x/std(x);
S = calcStats(x);
nSmp = size(S,2);
start = 1;
j=1;
while start < nSmp
    refStats = S(:,start);
    noChange = 1;
    i=1;
    while start + i <= nSmp && noChange
        noChange = ~any(thresholds<=abs(refStats-S(:,start+i)));
        i=i+1;
    end
    power(j) = mean(S(1,start:start+i-1));
    len(j)=i-1;
    j=j+1;
    start=start+i-1;
end
len = len(-30<power & 0.01<len/fs);
Lb=mean(len);
Rb=sum(len)/(nSmp-1);
psi=Lb*Rb/fs;

function S = calcStats(x)
N=length(x)-239-12;
S=zeros(13,N);
for i=1:N
   u=sum(x(i:i+239))/240;
   S(1,i)=10*log10(sum((x(i:i+239)-u).^2)/240);
   for j=1:12
       u=sum(x(i:i+239+j))/(240+j);
       x1=x(i:i+239)-u;
       x2=x(i+j:i+239+j)-u;
       S(j+1,i) = (x1'*x2)/(sqrt(x1'*x1)*sqrt(x2'*x2));
   end
end
```