On the Use of Asymmetric-shaped Tapers for Speaker Verification using I-vectors

Md Jahangir Alam, Patrick Kenny, Douglas O’Shaughnessy

Odyssey 2012

June 27, 2012
Commonly used features are MFCCs and PLP coefficients.
Introduction

- Commonly used features are MFCCs and PLP coefficients.

Purpose of windowing

Prior to the spectral analysis each frame is multiplied by a window (taper) to reduce the effect of discontinuity (at the beginning and end of a frame) introduced by the framing process.
Purpose of windowing

Due to windowing the peak is sharper and more distinct in the frequency response.
Purpose of windowing

Due to windowing the peak is sharper and more distinct in the frequency response.
Asymmetric Tapers (or windows)

- For short-time analysis of speech signals, most speaker or speech recognition systems use standard symmetric-shaped tapers such as Hamming or Hann.
- Symmetric tapers: Easy to implement and have linear phase property.
- Short-time (20-40 ms) phase spectrum plays very little (or, no) role in human perception tasks as well as in automatic speaker or speech recognition systems.
- Therefore, linearity constraint on phase can be removed without any adverse effects.
- Use of asymmetric tapers, having better magnitude response and shorter time delay [R. Rozman, Speech Comm. 2007], in speaker recognition can lead to a better recognition performance.
Asymmetric tapers (or windows) have been successfully applied in ASR tasks with improved performance compared to Hamming window [R. Rozman, Speech Comm. 2007, J.A. Morales-Cordovilla, CSSP (Springer) 2011 ]

Two asymmetric tapers have been incorporated in the MFCC features extraction framework for speaker recognition


\[
\begin{align*}
 w_{\text{ITU-TG.729}}(n) &= \begin{cases} 
 \frac{1+\alpha}{2} - \frac{1-\alpha}{2} \cos \left( \frac{2\pi n}{2N_L - 1} \right), & 0 \leq n \leq N_L - 1 \\
 \cos \left( \frac{2\pi (n-N_L)}{4N_R - 1} \right), & N_L \leq n \leq N - 1
\end{cases}
\end{align*}
\]

where \( N_L = \frac{5L}{6}, N_R = N - N_L, \alpha = 0.08. \)
Asymmetric tapers (or windows) in speaker recognition

2. **Asymmetric double dynamic range (DDR) Hamming window** [J.A. Morales-Cordovilla, CSSP (Springer) 2011, B. Shannon, Speech Comm. 2006]

\[
w_{\text{asymDDR}}(N - n + 1) = \begin{cases} 
    w_{\text{DDR}}(n), & (c - N/2) < n \leq (c + N/2) \\
    0, & \text{otherwise}
\end{cases}
\]

\(c\) is used to shift the peak position of \(w_{\text{DDR}}(n)\)
2. **Asymmetric double dynamic range (DDR) Hamming window** [J.A. Morales-Cordovilla, CSSP (Springer) 2011, B. Shannon, Speech Comm. 2006]

\[
    w_{\text{asymDDR}}(N-n+1) = \begin{cases} 
    w_{\text{ddr}}(n), & (c-N/2) < n \leq (c+N/2) \\
    0, & \text{otherwise}
    \end{cases}
\]

\(c\) is used to shift the peak position of \(w_{\text{ddr}}(n)\).
Asymmetric tapers (or windows) in speaker recognition
Asymmetric tapers (or windows) in speaker recognition
Asymmetric tapers (or windows) in speaker recognition

Introduction
Asymmetric tapers (or windows) in speaker recognition
Performance Evaluation
Conclusion

Asymmetric tapers (or windows) in speaker recognition

Original signal

Hamming windowed signal

asymDDR windowed signal

ITU-T G.729 windowed signal

Rectangular

Hamming

asymDDR

ITU-T G.729
Asymmetric tapers (or windows) in speaker recognition

- Hamming (clean)
- ITU-T G.729 (clean)
- asymDDR (clean)

Frequency
Time (s)
Asymmetric tapers (or windows) in speaker recognition

- Hamming (subway 5 dB)
- ITU-T G.729 (subway 5 dB)
- asymDDR (subway 5 dB)
Experimental Setup

- Trial lists: extended core-core condition of the 2010 NIST speaker recognition evaluation (SRE)

- We report results on det5 (telephone/telephone) condition obtained with the i-vector - Gaussian-PLDA system [Senoussaoui, M., Interspeech 2011, Kenny, P. Odyssey 2010, Dehak, N., IEEE TASLP 2011]

- For evaluating performances under additive noise distortion: subway noise, SNR = 5 dB

- Evaluation metrics: Equal error rate (EER), the "new DCF" (the detection cost function introduced in 2010) and the "old DCF"
Experimental Setup

I-vector extractor
Adding Noise

Noise is artificially added to the speech signal as follows:

- The active speech level of the filtered clean speech signal is first determined using the method of ITU-T P.56.
- A noise segment of the same length as the speech signal is randomly cut out of the noise recordings, appropriately scaled to reach the desired SNR level and finally added to the clean speech signal.
Experimental Setup

Adding Noise

Noise is artificially added to the speech signal as follows:

- The active speech level of the filtered clean speech signal is first determined using the method of ITU-T P.56.
- A noise segment of the same length as the speech signal is randomly cut out of the noise recordings, appropriately scaled to reach the desired SNR level and finally added to the clean speech signal.
Results

- Training on clean i-vectors, testing on clean i-vectors
- det5 (telephone/telephone)

For Female trials

<table>
<thead>
<tr>
<th></th>
<th>Hamming</th>
<th>ITU-T G.729</th>
<th>asymmetric DDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>EER</td>
<td>2.5%</td>
<td>2.15%</td>
<td>2.31%</td>
</tr>
<tr>
<td>old DCF</td>
<td>0.12</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>new DCF</td>
<td>0.39</td>
<td>0.35</td>
<td>0.38</td>
</tr>
</tbody>
</table>
Results

- Training on clean i-vectors, testing on clean i-vectors
- det5 (telephone/telephone)

For Female trials

<table>
<thead>
<tr>
<th></th>
<th>Hamming</th>
<th>ITU-T G.729</th>
<th>asymmetric DDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>EER</td>
<td>2.5%</td>
<td>2.15%</td>
<td>2.31%</td>
</tr>
<tr>
<td>old DCF</td>
<td>0.12</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>new DCF</td>
<td>0.39</td>
<td>0.35</td>
<td>0.38</td>
</tr>
</tbody>
</table>

For Male trials

<table>
<thead>
<tr>
<th></th>
<th>Hamming</th>
<th>ITU-T G.729</th>
<th>asymmetric DDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>EER</td>
<td>1.8%</td>
<td>2.02%</td>
<td>1.67%</td>
</tr>
<tr>
<td>old DCF</td>
<td>0.096</td>
<td>0.088</td>
<td>0.089</td>
</tr>
<tr>
<td>new DCF</td>
<td>0.32</td>
<td>0.29</td>
<td>0.31</td>
</tr>
</tbody>
</table>
## Results (Subway noise, SNR = 5 dB)

- Training on clean i-vectors, testing on noisy i-vectors
- det5 (telephone/telephone)

### For Female trials

<table>
<thead>
<tr>
<th></th>
<th>Hamming</th>
<th>ITU-T G.729</th>
<th>asymmetric DDR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EER</strong></td>
<td>6.29%</td>
<td>5.63%</td>
<td>6.29%</td>
</tr>
<tr>
<td><strong>old DCF</strong></td>
<td>0.32</td>
<td>0.29</td>
<td>0.31</td>
</tr>
<tr>
<td><strong>new DCF</strong></td>
<td>0.83</td>
<td>0.76</td>
<td>0.79</td>
</tr>
</tbody>
</table>
Results (Subway noise, SNR = 5 dB)

- Training on clean i-vectors, testing on noisy i-vectors
- det5 (telephone/telephone)

### For Female trials

<table>
<thead>
<tr>
<th></th>
<th>Hamming</th>
<th>ITU-T G.729</th>
<th>asymmetric DDR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EER</strong></td>
<td>6.29%</td>
<td>5.63%</td>
<td>6.29%</td>
</tr>
<tr>
<td><strong>old DCF</strong></td>
<td>0.32</td>
<td>0.29</td>
<td>0.31</td>
</tr>
<tr>
<td><strong>new DCF</strong></td>
<td>0.83</td>
<td>0.76</td>
<td>0.79</td>
</tr>
</tbody>
</table>

### For Male trials

<table>
<thead>
<tr>
<th></th>
<th>Hamming</th>
<th>ITU-T G.729</th>
<th>asymmetric DDR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EER</strong></td>
<td>6.73%</td>
<td>7.04%</td>
<td>6.80%</td>
</tr>
<tr>
<td><strong>old DCF</strong></td>
<td>0.32</td>
<td>0.35</td>
<td>0.34</td>
</tr>
<tr>
<td><strong>new DCF</strong></td>
<td>0.79</td>
<td>0.81</td>
<td><strong>0.79</strong></td>
</tr>
</tbody>
</table>
Generalized method for construction of an asymmetric window from a symmetric window function

- Performed well in ASR task [Alam, J., EUSIPCO 2012].
Generalized method for asymmetric window

Generalized method for construction of an asymmetric window from a symmetric window function

- Performed well in ASR task [Alam, J., EUSIPCO 2012].

\[
w_{\text{asy}(n)} = a(n) w_{\text{sym}(n)} e^{\kappa \theta(n)}, \quad 0 \leq n \leq N - 1
\]
Conclusion

- Two asymmetric windows have been incorporated in the MFCCs computation framework for speaker verification task.
- Replacing a window function, is a simple procedure, does not increase the system complexity.
- Asymmetric windows outperformed the symmetric Hamming window (in most det conditions) both in clean and additive noise environments.
- There should be a trade-off between mainlobe width and the height of sidelobes.
- Window centered on the average pitch helps to have a good characterization of the formants [J.A. Morales-Cordovilla, Circuit Syst. Signal Process (CSSP). 2011]
Questions and/or Suggestions?