One-class Learning Towards Synthetic Voice Spoofing Detection

You Zhang, Fei Jiang, Zhiyao Duan

University of Rochester, USA

Outline

Background  Method  Experiments  Conclusion
Clone a Voice in Five Seconds With This AI Toolbox

A new Github project introduces a remarkable Real-Time Voice Cloning Toolbox that enables anyone to clone a voice from as little as five seconds of sample audio.

I trained an AI to copy my voice and it scared me silly

by ABHIMANYU GHOSHAL — Jan 22, 2018 in INSIGHTS

Hey Google, turn on the Christmas tree.

THE WALL STREET JOURNAL.

Fraudsters Used AI to Mimic CEO’s Voice in Unusual Cybercrime Case

Scams using artificial intelligence are a new challenge for companies
Anti-spoofing

• Spoofing Countermeasure: Detect spoofing attacks

Impersonation / Replay Spoofing attack

Text-to-speech / Voice conversion Spoofing attack

Sensor (microphone) → Feature Extraction

Voice Anti-spoofing

Speaker Verification

Decision Accept or Reject

Impersonation / Replay Spoofing attack

Text-to-speech / Voice conversion Spoofing attack

Sensor (microphone)

speech

features
Research question

Motivation:

• The fast development of speech synthesis are posing increasingly more threat.

• The distribution mismatch between the training set and test set for the spoofing attacks class.

➢ How can the anti-spoofing system defend against unseen spoofing attacks?

(Generalization ability)
Outline

Introduction  Method  Experiments  Conclusion
Definition of one-class

• “In one-class classification, one of the classes (referred to as the positive class or target class) is well characterized by instances in the training data. For the other class (nontarget), it has either no instances at all, very few of them, or they do not form a statistically-representative sample of the negative concept.”

One-class learning (OC-Softmax)

(a) Original Softmax  (b) AM-Softmax  (c) OC-Softmax (Proposed)

Fig. 1. Illustration of the original Softmax and AM-Softmax for binary classification, and our proposed OC-Softmax for one-class learning. (The embeddings and the weight vectors shown are non-normalized).
One-Class Softmax (Proposed)

- **Training (Loss):**

\[ \mathcal{L}_{OCS} = \frac{1}{N} \sum_{i=1}^{N} \log \left( 1 + e^{\alpha(y_i \hat{w}_0 \hat{x}_i)(-1)^{y_i}} \right). \]

- **Inference (Score):**

\[ S_{OCS} = \hat{w}_0 \hat{x}_i. \]
Comparing loss

- **Softmax**

\[
\mathcal{L}_S = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{w^T y_i x_i}}{e^{w^T y_i x_i} + e^{w^T (1-y_i) x_i}}
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} \log \left( 1 + e^{(w_{1-y_i} - w_{y_i})^T x_i} \right),
\]

- **AM-Softmax**

\[
\mathcal{L}_{AMS} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{\alpha (\hat{w}_{y_i}^T \hat{x}_i - m)}}{e^{\alpha (\hat{w}_{y_i}^T \hat{x}_i - m)} + e^{\alpha \hat{w}_{1-y_i}^T \hat{x}_i}}
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} \log \left( 1 + e^{\alpha (m - (\hat{w}_{y_i} - \hat{w}_{1-y_i})^T \hat{x}_i)} \right),
\]

- **OC-Softmax**

\[
\mathcal{L}_{OCS} = \frac{1}{N} \sum_{i=1}^{N} \log \left( 1 + e^{\alpha (m_{y_i} - \hat{w}_0 \hat{x}_i)(-1)^{y_i}} \right).
\]
Outline

Introduction
Method
Experiments
Conclusion
Dataset

ASVspoof 2019 Logical Access (TTS + VC)

- Bona fide speech (VCTK dataset)
- 6 known attacks (appear in training set)
- 13 unknown attacks (only appear in eval set)

<table>
<thead>
<tr>
<th>Bona fide</th>
<th>Spoofed</th>
</tr>
</thead>
<tbody>
<tr>
<td># utterance</td>
<td># utterance</td>
</tr>
<tr>
<td>Training</td>
<td>2,580</td>
</tr>
<tr>
<td>Development</td>
<td>2,548</td>
</tr>
<tr>
<td>Evaluation</td>
<td>7,533</td>
</tr>
</tbody>
</table>
Evaluation of OC-Softmax

• Results on the development and evaluation sets of the ASVspoof 2019 LA scenario using different losses

<table>
<thead>
<tr>
<th>Loss</th>
<th>Dev Set</th>
<th>Eval Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER (%)</td>
<td>t-DCF</td>
</tr>
<tr>
<td>Softmax</td>
<td>0.35</td>
<td>0.010</td>
</tr>
<tr>
<td>AM-Softmax</td>
<td>0.43</td>
<td>0.013</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.20</td>
<td>0.006</td>
</tr>
</tbody>
</table>

• OC-Softmax performs the best on unseen attacks.

Feature Embedding Visualization
## Comparison with single systems

<table>
<thead>
<tr>
<th>System</th>
<th>EER (%)</th>
<th>min t-DCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQCC + GMM [3]</td>
<td>9.57</td>
<td>0.237</td>
</tr>
<tr>
<td>LFCC + GMM [3]</td>
<td>8.09</td>
<td>0.212</td>
</tr>
<tr>
<td>Chettri et al. [22]</td>
<td>7.66</td>
<td>0.179</td>
</tr>
<tr>
<td>Monterio et al. [14]</td>
<td>6.38</td>
<td>0.142</td>
</tr>
<tr>
<td>Gomez-Alanis et al. [16]</td>
<td>6.28</td>
<td>-</td>
</tr>
<tr>
<td>Aravind et al. [18]</td>
<td>5.32</td>
<td>0.151</td>
</tr>
<tr>
<td>Lavrentyeva et al. [21]</td>
<td>4.53</td>
<td>0.103</td>
</tr>
<tr>
<td>ResNet + OC-SVM</td>
<td>4.44</td>
<td>0.115</td>
</tr>
<tr>
<td>Wu et al. [17]</td>
<td>4.07</td>
<td>0.102</td>
</tr>
<tr>
<td>Tak et al. [19]</td>
<td>3.50</td>
<td>0.090</td>
</tr>
<tr>
<td>Chen et al. [15]</td>
<td>3.49</td>
<td>0.092</td>
</tr>
<tr>
<td><strong>Proposed</strong></td>
<td><strong>2.19</strong></td>
<td><strong>0.059</strong></td>
</tr>
</tbody>
</table>
Results in the leader board

<table>
<thead>
<tr>
<th>#</th>
<th>ID</th>
<th>t-DCF</th>
<th>EER</th>
<th>#</th>
<th>ID</th>
<th>t-DCF</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T05</td>
<td>0.0809</td>
<td>0.22</td>
<td>26</td>
<td>T57</td>
<td>0.2059</td>
<td>10.65</td>
</tr>
<tr>
<td>2</td>
<td>T45</td>
<td>0.0510</td>
<td>1.86</td>
<td>27</td>
<td>T42</td>
<td>0.2080</td>
<td>8.01</td>
</tr>
<tr>
<td>3</td>
<td>T60</td>
<td>0.0755</td>
<td>2.64</td>
<td>28</td>
<td>B02</td>
<td>0.2116</td>
<td>8.09</td>
</tr>
<tr>
<td>4</td>
<td>T24</td>
<td>0.0953</td>
<td>3.45</td>
<td>29</td>
<td>T17</td>
<td>0.2129</td>
<td>7.63</td>
</tr>
<tr>
<td>5</td>
<td>T50</td>
<td>0.1118</td>
<td>3.56</td>
<td>30</td>
<td>T23</td>
<td>0.2180</td>
<td>8.27</td>
</tr>
<tr>
<td>6</td>
<td>T41</td>
<td>0.1131</td>
<td>4.50</td>
<td>31</td>
<td>T53</td>
<td>0.2252</td>
<td>8.20</td>
</tr>
<tr>
<td>7</td>
<td>T39</td>
<td>0.1203</td>
<td>7.42</td>
<td>32</td>
<td>T59</td>
<td>0.2298</td>
<td>7.95</td>
</tr>
<tr>
<td>8</td>
<td>T32</td>
<td>0.1239</td>
<td>4.92</td>
<td>33</td>
<td>B01</td>
<td>0.2366</td>
<td>9.57</td>
</tr>
<tr>
<td>9</td>
<td>T58</td>
<td>0.1333</td>
<td>6.14</td>
<td>34</td>
<td>T52</td>
<td>0.2366</td>
<td>9.25</td>
</tr>
<tr>
<td>10</td>
<td>T04</td>
<td>0.1404</td>
<td>5.74</td>
<td>35</td>
<td>T40</td>
<td>0.2417</td>
<td>8.82</td>
</tr>
<tr>
<td>11</td>
<td>T01</td>
<td>0.1409</td>
<td>6.01</td>
<td>36</td>
<td>T55</td>
<td>0.2681</td>
<td>10.88</td>
</tr>
<tr>
<td>12</td>
<td>T22</td>
<td>0.1545</td>
<td>6.20</td>
<td>37</td>
<td>T43</td>
<td>0.2720</td>
<td>13.35</td>
</tr>
<tr>
<td>13</td>
<td>T02</td>
<td>0.1552</td>
<td>6.34</td>
<td>38</td>
<td>T31</td>
<td>0.2788</td>
<td>15.11</td>
</tr>
<tr>
<td>14</td>
<td>T44</td>
<td>0.1554</td>
<td>6.70</td>
<td>39</td>
<td>T25</td>
<td>0.3025</td>
<td>23.21</td>
</tr>
<tr>
<td>15</td>
<td>T16</td>
<td>0.1569</td>
<td>6.02</td>
<td>40</td>
<td>T26</td>
<td>0.3036</td>
<td>15.09</td>
</tr>
<tr>
<td>16</td>
<td>T08</td>
<td>0.1583</td>
<td>6.38</td>
<td>41</td>
<td>T47</td>
<td>0.3049</td>
<td>18.34</td>
</tr>
<tr>
<td>17</td>
<td>T62</td>
<td>0.1628</td>
<td>6.74</td>
<td>42</td>
<td>T46</td>
<td>0.3214</td>
<td>12.59</td>
</tr>
<tr>
<td>18</td>
<td>T27</td>
<td>0.1648</td>
<td>6.84</td>
<td>43</td>
<td>T21</td>
<td>0.3393</td>
<td>19.01</td>
</tr>
<tr>
<td>19</td>
<td>T29</td>
<td>0.1677</td>
<td>6.76</td>
<td>44</td>
<td>T61</td>
<td>0.3437</td>
<td>15.66</td>
</tr>
<tr>
<td>20</td>
<td>T13</td>
<td>0.1778</td>
<td>6.57</td>
<td>45</td>
<td>T11</td>
<td>0.3742</td>
<td>18.15</td>
</tr>
<tr>
<td>21</td>
<td>T48</td>
<td>0.1791</td>
<td>9.08</td>
<td>46</td>
<td>T56</td>
<td>0.3856</td>
<td>15.32</td>
</tr>
<tr>
<td>22</td>
<td>T10</td>
<td>0.1829</td>
<td>6.81</td>
<td>47</td>
<td>T12</td>
<td>0.4088</td>
<td>18.27</td>
</tr>
<tr>
<td>23</td>
<td>T54</td>
<td>0.1852</td>
<td>7.71</td>
<td>48</td>
<td>T14</td>
<td>0.4143</td>
<td>20.60</td>
</tr>
<tr>
<td>24</td>
<td>T38</td>
<td>0.1940</td>
<td>7.51</td>
<td>49</td>
<td>T20</td>
<td>1.0000</td>
<td>92.36</td>
</tr>
<tr>
<td>25</td>
<td>T33</td>
<td>0.1960</td>
<td>8.93</td>
<td>50</td>
<td>T30</td>
<td>1.0000</td>
<td>49.60</td>
</tr>
</tbody>
</table>

Ours 0.059 2.19

- Could rank between the 2\textsuperscript{nd} and 3\textsuperscript{rd}
- Top systems all use model fusion, but we do not
Takeaways

• One-class learning aims to **compact the target** class representation in the embedding space, set a tight classification boundary around it and **push away non-target**.

• The proposed OC-Softmax could improve the **generalization ability** of anti-spoofing system against **unseen spoofing attacks**.
Follow-up works

• Channel Robustness

• Joint Optimization with ASV
Future directions

• Defend against diversified spoofing attacks
  o TTS+VC, replay
  o Partially spoofed
  o Adversarial attack

• Explainable anti-spoofing
  o Understanding the difference between synthetic vs. natural speech

• Visually-informed anti-spoofing
  o Deepfake detection, multimedia forensics
Thank you!

Q & A
Takeaways

• One-class learning aims to **compact the target** class representation in the embedding space, set a tight classification boundary around it and **push away non-target**.

• One-class learning could improve the **generalization ability** of anti-spoofing system against **unknown spoofing attacks**.
Voice Biometrics

• Speaker Verification: Verify the identity of a speaker
Spoofing attacks

- **Impersonation**
  -- twins and professional mimics, no database available

- **Replay**
  -- reuse pre-recorded audio, most accessible

- **Text-to-speech (TTS)**
  -- convert written text into spoken words with speech synthesis

- **Voice conversion (VC)**
  -- convert speech from source speaker to target speaker’s voice
ASVspoof Challenge

• Logical access (LA)  \{ Text-to-speech (TTS)  \\
  Voice conversion (VC)  \\
  TTS+VC  \\
  -- algorithm-related artifacts  ★ our current focus

• Physical access (PA) -- pre-recorded, replay  \\
  -- device-related artifacts
Binary versus One-Class Classification

(a) Binary classification

(b) One-class classification
Softmax

• Training (Loss):

\[ \mathcal{L}_S = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{w_{y_i}^T x_i}}{e^{w_{y_i}^T x_i} + e^{w_{1-y_i}^T x_i}} \]

\[ = \frac{1}{N} \sum_{i=1}^{N} \log (1 + e^{(w_{1-y_i} - w_{y_i})^T x_i}) , \]

• Inference (Score):

\[ S_S = \frac{e^{w_0^T x_i}}{e^{w_0^T x_i} + e^{w_1^T x_i}} . \]
Additive Margin Softmax

- Training (Loss):

\[ \mathcal{L}_{AMS} = - \frac{1}{N} \sum_{i=1}^{N} \log \left( \frac{e^{\alpha (\hat{w}_{y_i}^T \hat{\mathbf{x}}_i - m)}}{e^{\alpha (\hat{w}_{y_i}^T \hat{\mathbf{x}}_i - m)} + e^{\alpha \hat{w}_{1-y_i}^T \hat{\mathbf{x}}_i}} \right) \]

\[ = \frac{1}{N} \sum_{i=1}^{N} \log \left( 1 + e^{\alpha (m - (\hat{w}_{y_i} - \hat{w}_{1-y_i})^T \hat{\mathbf{x}}_i)} \right), \]

- Inference (Score):

\[ S_{AMS} = (\hat{w}_0 - \hat{w}_1)^T \hat{\mathbf{x}}_i. \]
OC-Softmax output as probability

\[
L_{OC} = \frac{1}{N} \sum_{i=1}^{N} \log \left( 1 + e^{\alpha (m_{yi} - \hat{w}^T \hat{x}_i)(-1)^{y_i}} \right)
\]

\[
= \frac{1}{N} \left( \sum_{|\Omega|} \log \left( 1 + e^{\alpha (m_{0} - \hat{w}^T \hat{x}_i)} \right) + \sum_{|\Omega|} \log \left( 1 + e^{\alpha (\hat{w}^T \hat{x}_i - m_{1})} \right) \right)
\]

\[
= -\frac{1}{N} \left( \sum_{|\Omega|} \log \frac{1}{1 + e^{\alpha (m_{0} - \hat{w}^T \hat{x}_i)}} + \sum_{|\Omega|} \log \frac{1}{1 + e^{\alpha (\hat{w}^T \hat{x}_i - m_{1})}} \right)
\]