Active Learning for Speech Emotion Recognition using Deep Neural Network

Mohammed Abdelwahab And Carlos Busso
Generalization of Models

- Mismatch between train and test conditions is one of the main barriers in speech emotion recognition.
- **Under ideal classification conditions**
  - The training and testing sets come from the same domain.
- **Under real application conditions**
  - The training and testing sets come from different domains.
  - This leads to performance drop [Shami and Verhelst 2007, Parthasarathy and Busso 2017].

<table>
<thead>
<tr>
<th>Training</th>
<th>Testing</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danish</td>
<td>Danish</td>
<td>64.90 %</td>
</tr>
<tr>
<td>Berlin</td>
<td>Berlin</td>
<td>80.70 %</td>
</tr>
<tr>
<td>Berlin</td>
<td>Danish</td>
<td>22.90 %</td>
</tr>
<tr>
<td>Danish</td>
<td>Berlin</td>
<td>52.60 %</td>
</tr>
</tbody>
</table>

[Shami and Verhelst 2007]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>In-corpus</td>
<td>0.764</td>
<td>0.289</td>
<td>0.713</td>
</tr>
<tr>
<td>Cross-corpus</td>
<td>0.464</td>
<td>0.184</td>
<td>0.451</td>
</tr>
</tbody>
</table>

[Parthasarathy and Busso 2017]
The performance of a classifier degrades if there is a mismatch between training and testing conditions

- Speaker variations, channels (environments, noise), language, and microphone settings

How to build a classifier that generalizes well?

- Minimize the discrepancy between the source and target domains

We explore this problem relying on active learning
Active learning has been widely used to iteratively select training samples that maximizes the model’s performance
- Not all the samples are equal

DNN pushes state of the art performance
- It requires vast amounts of labeled data

There is a need for scalable active learning approach for DNN
- Explore the approaches to identify most useful $N$ samples
Related Work

- **Speech Emotion Recognition**
  - Use labelers’ agreement to build uncertainty models [Zhang et. al. 2013]
  - Multi-view uncertainty sampling to minimize amount of labeled data [Zhang et. al. 2015]
  - Minimize annotations per sample using agreement threshold [Zhang et. al. 2015]
  - Minimize noise accumulation in self-training [Zhang et. al. 2016]
  - Adapt model with low confidence correctly classified samples [Abdelwahab & Busso 2017]
  - Combine Ensembles and Active learning to mitigate performance loss in new domain [Abdelwahab & Busso 2017]
  - Greedy sampling for Multi-task speech emotion linear regression [Wu & Huang 2018]

- **None of those approaches used Deep Neural networks**
There is no data acquisitions functions that work well in all scenarios

- **Heuristic approaches where shown to work in practice**
  - **Greedy sampling**
    - Label space
    - Feature space
    - Combination
  - **Uncertainty sampling**
    - Least confident samples
    - Margin
    - Entropy
    - Vote Entropy (Ensembles)
  - Dropout
  - Random sampling (baseline)
Greedy Sampling Approach

- **Greedy sampling for regression** [Wu et al., 2019]
  - maximize the diversity in the train set
  1. Select initial samples
    - Previously selected samples
  2. Compute distances
    - Features space
      \[ d_x^{i,j} = \| x_i - x_j \|_2 \]
    - Label space
      \[ d_y^{i,j} = |\hat{y}_i - y_j| \]
    - Combination
      \[ d_{xy}^{i,j} = d_x^{i,j} d_y^{i,j} \]
  3. Select \( k \) samples to annotate
  4. Update model and repeat
Uncertainty Sampling: Dropout

- **Dropout can approximate Bayesian inference** [Gal et al., 2016]
  - We can represent the models’ uncertainty
  - Use different configurations of dropout, analyzing predictions per sample

- **Goal:** select samples that the existing model is the most uncertain across several dropout iterations
- Use existing podcast recordings
- Divide into speaker turns
- Emotion retrieval to balance the emotional content
- Annotate using crowdsourcing framework

Podcast recording
The MSP-Podcast Database

**MSP-Podcast**

- Collection of publicly available podcasts (naturalness and the diversity of emotions)
  - Interviews, talk shows, news, discussions, education, storytelling, comedy, science, technology, politics, etc.
- Creative Commons copyright licenses
- Single speaker segments, High SNR, no music, no phone quality
- Developing and optimizing different machine learning framework using existing databases
  - Balance the emotional content
- Emotional annotation using crowdsourcing platform
MSP-Podcast corpus version 1.1

Segmented turns 152,975 sentences over 1,000 podcasts

With emotion labels: 22,630 sentences (38h, 57m)

- Test set
  - 7,181 segments from 50 speakers (25 males, 25 females)
- Development set
  - 2,614 segments from 15 speakers (10 males, 5 females)
- Train set
  - remaining 12,830 segments

Arousal

Valence

Dominance
Acoustic Features

- **Interspeech 2013 Feature set**
  - 65 low level descriptors (LLD)
  - Functional are calculated on LLDs resulting in total of 6,373 features
- Functionals include:
  - Quartile ranges
  - Arithmetic mean
  - Root quadratic mean
  - Moments
  - Mean/std. of rising/ falling slopes

<table>
<thead>
<tr>
<th>4 energy related LLD</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum of auditory spectrum (loudness)</td>
<td>prosodic</td>
</tr>
<tr>
<td>Sum of RASTA-filtered auditory spectrum</td>
<td>prosodic</td>
</tr>
<tr>
<td>RMS Energy, Zero-Crossing Rate</td>
<td>prosodic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>55 spectral LLD</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>RASTA-filt. aud. spect. bds. 1–26 (0–8 kHz)</td>
<td>spectral</td>
</tr>
<tr>
<td>MFCC 1–14</td>
<td>cepstral</td>
</tr>
<tr>
<td>Spectral energy 250–650 Hz, 1 k–4 kHz</td>
<td>spectral</td>
</tr>
<tr>
<td>Spectral Roll-Off Pt. 0.25, 0.5, 0.75, 0.9</td>
<td>spectral</td>
</tr>
<tr>
<td>Spectral Flux, Centroid, Entropy, Slope</td>
<td>spectral</td>
</tr>
<tr>
<td>Psychoacoustic Sharpness, Harmonicity</td>
<td>spectral</td>
</tr>
<tr>
<td>Spectral Variance, Skewness, Kurtosis</td>
<td>spectral</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6 voicing related LLD</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_0$ (SHS &amp; Viterbi smoothing)</td>
<td>prosodic</td>
</tr>
<tr>
<td>Prob. of voicing</td>
<td>voice qual.</td>
</tr>
<tr>
<td>log. HNR, Jitter (local &amp; $\delta$), Shimmer (local)</td>
<td>voice qual.</td>
</tr>
</tbody>
</table>
**Proposed Architecture**

- **Multitask learning network:**
  - Primary task: emotion regression
    - concordance correlation coefficient (CCC)
  - Secondary task: feature reconstruction
    - Mean square error (MSE)

\[ L = \frac{1}{N} \sum_{i=1}^{N} \| x - \hat{x} \|^2 + \lambda_2 \left[ 1 - \frac{2\rho \sigma_y \sigma_{\hat{y}}}{\sigma_y^2 + \sigma_{\hat{y}}^2 + (\mu_{\hat{y}} - \mu_y)^2} \right] \]

- Secondary task helps to generalize the model, especially with limited data

![Diagram of the proposed architecture](attachment:architecture.png)
Experimental Settings

- We consider 50, 100, 200, 400, 800, and 1200 samples
- Samples are selected based on the latest model
- We consider two starting points
  - From scratch
  - Autoencoder trained on reconstruction loss only for 20 epochs
- Results are the average of 20 trials
- Greedy sampling (feature space):
  - Use embedding of the autoencoder to reduce the search space
Observation
- Greedy feature leads to better performance
- Dropout is not as effective
- Random approach best methods as we add more data
- Pretrained encoder helps with limited samples
- **Observations**
  - Pretrained autoencoder helps to achieve better performance
  - We approach within corpus performance with only 10% of the training data
  - Random sampling is less effective
## Statistical Significance

<table>
<thead>
<tr>
<th># samples</th>
<th>Arousal</th>
<th>Valence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>Random Sampling</td>
<td>0.57</td>
<td>0.61</td>
</tr>
<tr>
<td>Greedy Feature</td>
<td><strong>0.58</strong></td>
<td><strong>0.64</strong></td>
</tr>
<tr>
<td>Greedy Label</td>
<td>0.50</td>
<td>0.53</td>
</tr>
<tr>
<td>Greedy Combination</td>
<td>0.52</td>
<td>0.55</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.56</td>
<td>0.59</td>
</tr>
</tbody>
</table>

Bold: statistically significant improvements over random sampling

### Observations
- Greedy sampling in feature space almost always better than random sampling
- Dropout was not as effective
Sensitivity to $k$ (how often we update the model)

- Multitask autoencoder framework with greedy methods
  - No statistical difference with $k = 1$ and $k = 10$
  - Method is not sensitive to this parameter (reduce complexity)
Consistency of the Results

- **20 results starting with different initializations**
  - Greedy sampling on the feature space versus random sampling
  - Standard deviation of the CCC values achieved by the greedy sampling method decreases faster as the sampling size increases
    - More consistent than random sampling
Conclusions

- Greedy sampling achieves higher performance with lower variance compared to random sampling
- Greedy sampling in label space depends on model’s performance
- As we introduced more data, the differences in performance across data acquisition functions reduce
- Reduce computation cost:
  - Calculate the distance in embedding with lower dimensions
  - Set adequate value of $k$ reduces the frequency of model updates

- Future Work
  - Combine active learning with curriculum learning
  - Consider new acquisition functions that scale well
This work was funded by NSF CAREER award IIS-1453781

Interested on our research?
msp.utdallas.edu