Building language detectors with small amounts of data

David van Leeuwen (TNO)
Niko Brümmer (SDV)
Synopsis

- Standard language model training for language recognition needs lots of data
  - typically 60 hours speech, 100 speakers, per language

- We would like to reduce this demand

- Investigate classifier that works in score space rather than acoustic space

- Evaluate with
  - LRE-2005 (7 languages)
  - CSLU-22 (21 languages)

- Can train score-space based system with ~ 1 hour data
  - at twice the $C_{DET}$
Motivation

- Sometimes there is not much training data for new language available
  - e.g., Indian accented English in LRE-2005: 20 minutes

- Sometimes we may not want to train acoustic model for new language
  - Hard for inexperienced user

- Notion that language recognition back-end can ‘repair’ sub-optimal modeling performance
  - Try to let back-end to the whole job, without specific acoustic language model
Caveat

- Collection of large amounts of speech *should* be relatively easy
  - no orthographic annotation required

- But:
  - correct labeling of language *is* required
  - different collection characteristic to background data will lead to confounding of language and data collection modeling

- This is true for *any* kind of modeling
  - front-end (GMM, SVM, acoustics, phonotactics)
  - back-end (LDA, logistic regression)
Overview of (typical) LID system

SLD: Single Language Detector

Gaussian back end = LDA + calibration

$N_L$ posteriors
Modeling power of LDA back-end

- with proper priors and threshold for posterior
  - optimal NIST LRE decisions can be made

- LRE-2005 languages
  \[ p(L_{2005}) = \frac{1}{N_L} \]

- Other CallFriend
  \[ p(L_{CF\ 2005}) = 0 \]

- Posterior threshold
  \[ \theta = \frac{1}{N_L} \]
LRE-2005: Jack-knifing approach

• for each target language $L_i$
  • remove Single Language Detector $L_i$ from LDA training
  • build LDA, using all LDA training trials (incl. $L_i$)
  • compute target and non-target scores for these $L_i$ test-segments, and make decisions
• pool decisions, calculate $C_{DET}$ according to NIST LRE plan

LDA: Linear Discriminant Analysis
$C_{DET}$: Cost of detection
LRE: Language Recognition Evaluation
Application to NIST LRE-2005

CallFriend training
12 languages

LDA training trials

Linear Discriminant Analysis training

Single Language Detectors

LDA Back-end

Test trials

Score
Application to NIST LRE-2005

CallFriend training
12 languages

LDA training trials

Linear Discriminant Analysis training

Test trials

Score

Single Language Detectors

LDA Back-end
Application to NIST LRE-2005

CallFriend training
12 languages

LDA training trials

Linear Discriminant Analysis training

Test trials

Single Language Detectors

LDA Back-end

Score
Three systems: 1) Chan-GMM

- Channels: ch1, ch2, ch3, ch4, ch5, ch6
- UBMs
- Languages
- GMMs

Likelihood ratios: $N_{ch} \times N_L$
Three systems: 2) GMS (GMM means SVM)

- s1
- s2

Languages

Sexes

GMM means

+ 

− 

SVM

2 \( N_L \) Scores
Three systems: 3) Chan-GMS

- **Channels**: ch1, ch2, ch3, ch4, ch5, ch6
- **Languages**: SVM
- **UBMs**: +, −
- **GMM means**: N_ch × N_L
- **Scores**: SVM
Results: Sparse Training Performance

- 30–60 hours per language for SLD
- 1.9–7.6 hours per language for LDA
  - collection of NIST trial sets ’96–’03
- Observations
  - GMS best baseline
  - Chan-GMM most robust
  - Chan-GMS best sparse training
Results: Effect of number of Single Language Detectors

- ‘Columns’ in LDA matrix
- random selection of $r$ columns per language
  - $r = 1\ldots 6$
  - average 10 runs
- Chan-GMS system
- sparse training constant hit
Results: effect of sparse training size

- ‘Rows’ in LDA matrix
- Fraction of LDA training trials retained
  - $2^{-5} \ldots 2^{0}$
  - random selection
  - average 10 runs
- GMS
  - large hit sparse
  - less hit by training size
- Chan-GMS
  - smaller hit sparse
  - more hit by LDA
Final test: independent data collection

- Use CSLU22 data collection as independent test
  - 21 languages
  - 2000+ speakers
  - Superset of LRE-2005 languages
  - ‘story’ sentences, 37s mean duration
  - 10-fold cross validation LDA-train / test

- ~ 54 min LDA training / language

- Full CallFriend training for SLDs
Results CSLU22 per language

CallFriend languages

New languages

C_{DET} (%)
Results CSLU: in/out set SLDs

<table>
<thead>
<tr>
<th>CDET (%)</th>
<th>mean CallFriend</th>
<th>mean New</th>
<th>LRE-05 languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6.2</td>
<td>5.2</td>
<td>4.6</td>
</tr>
<tr>
<td>5</td>
<td>15.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>10.6</td>
<td>4.5</td>
</tr>
<tr>
<td>15</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

© 2008 David van Leeuwen
Conclusions

• LDA can model new language for LID quite efficiently
  • very fast training of LDA
  • ~ 1 hour of training data gives $C_{\text{DET}}$ within factor ~ 2

• Generative GMMs seems more robust for missing SLD
  • but baseline performance is worse than discriminative GMS
  • rely more on back-end, anyway

• More SLDs in LDA
  • make LDA more robust for new language missing in SLDs
  • need more training data for LDA
    • including new language

• Discriminative channel-dependent GMS trade-off between
  • good baseline performance
  • fair robustness for language missing from SLDs