Rapid Adaptation for Mobile Speech Applications

Michiel Bacchiani

Google

michiel@google.com

Introduction

- Mobile applications see a large population of speakers in many environments.
- Interactions are very short making adaptation challenging.
- Data sparsity make MAP or linear transforms-based adaptation infeasible.
- Data pooling is an option to address sparsity but
  - Need user approval for use.
  - Personal data still from many environments.
  - Pooling makes the application much more complex.

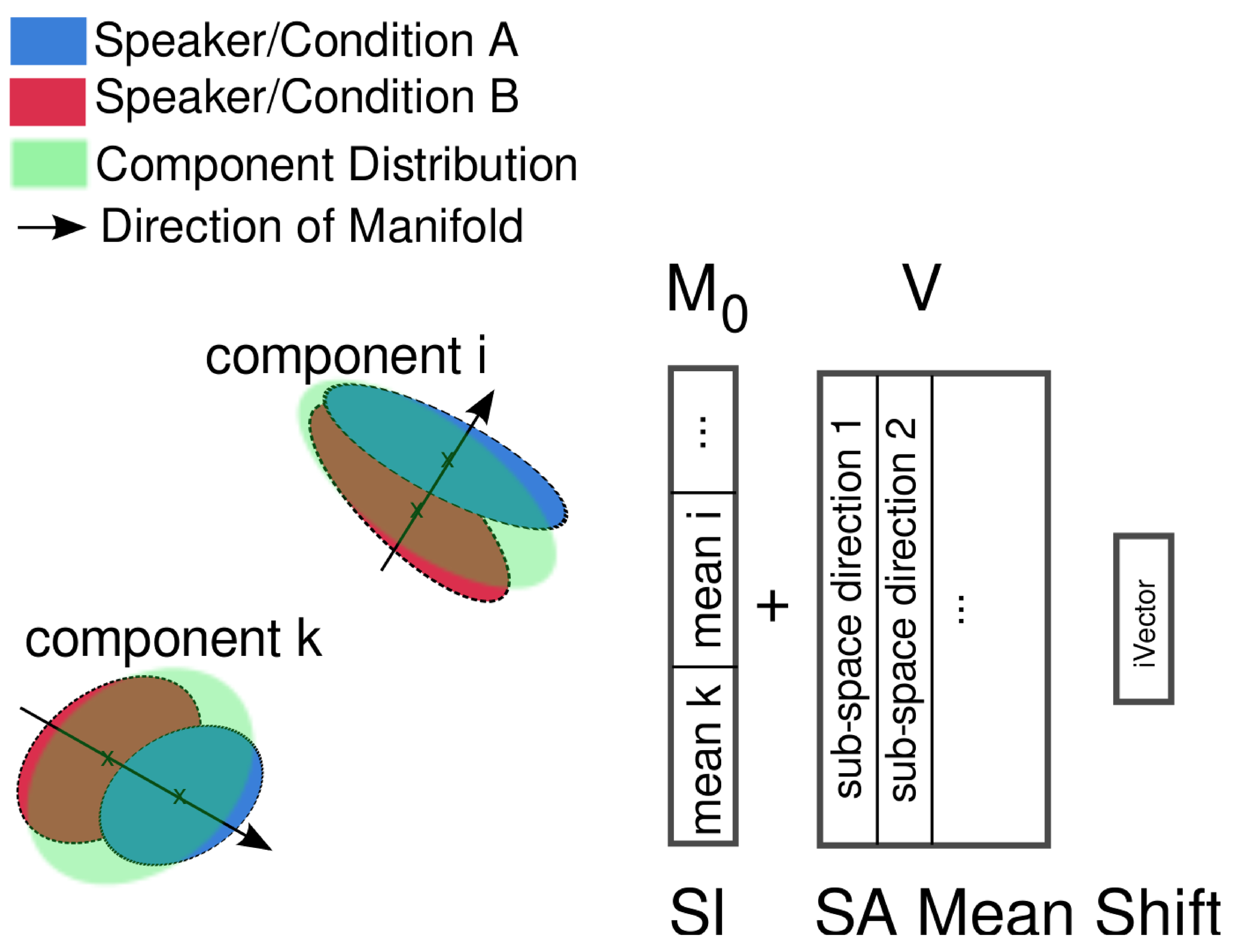
Rapid Adaptation Techniques

- Pilot experiment using data pooling of opt-in user data showed an error rate reduction of only 6% using linear transform-based adaptation. Likely due to diversity in the recording conditions.
- Focus on rapid adaptation techniques that factor the model in training and use a parsimonious sub-space location for estimation of the adapted model.
- Rapid adaptation and related techniques:
  - **Eigenvoices:** Pool training data by speaker/recording condition to define model sub-space bases in the form of GMMs.
  - **CAT:** Estimate a set of MLLR transforms on the training data that define sub-space bases.
  - **iVectors:** Popular in speaker identification. Define sub-space bases as Text-Independent GMMs. Define speaker/condition by sub-space location estimation on a per-utterance basis.

  In this work we focus on the iVector approach of estimating per-utterance sub-space dimensions but using, like in Eigenvoices, recognition GMMs as the bases.

  - Larger training data fragmentation than for eigenvoices (per utterance as opposed to per speaker).
  - Larger sub-space bases than for iVectors (recognition model as opposed to text independent GMM).
  - Larger model and larger fragmentation complicate parameter estimation, perturb and estimate bases in stages.

iVector Model



- Sub-space model:

$$M(i) = M_0 + V y(i)$$

- Posterior distribution for iVectors $y(i)$ is Gaussian with covariance and mean:

$$l(i) = I + V^T \Sigma^{-1} N^i V$$
$$a(i) = l(i)^{-1} V^T \Sigma^{-1} S^i$$

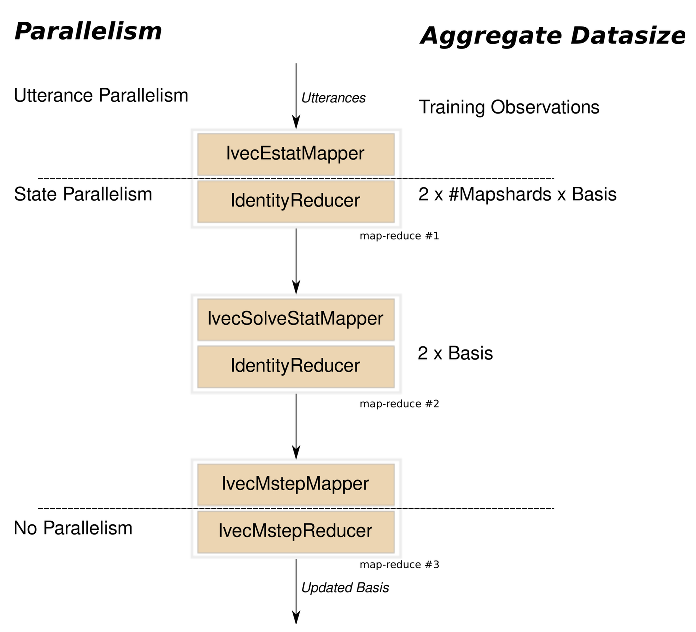
- EM-based training uses iVector expectations:

$$\mathrm{E}\left[y(i)\right] = a(i)$$
$$\mathrm{E}\left[y(i)y(i)^T\right] = \mathrm{E}\left[y(i)\right]\mathrm{E}\left[y(i)^T\right] + l(i)^{-1}$$

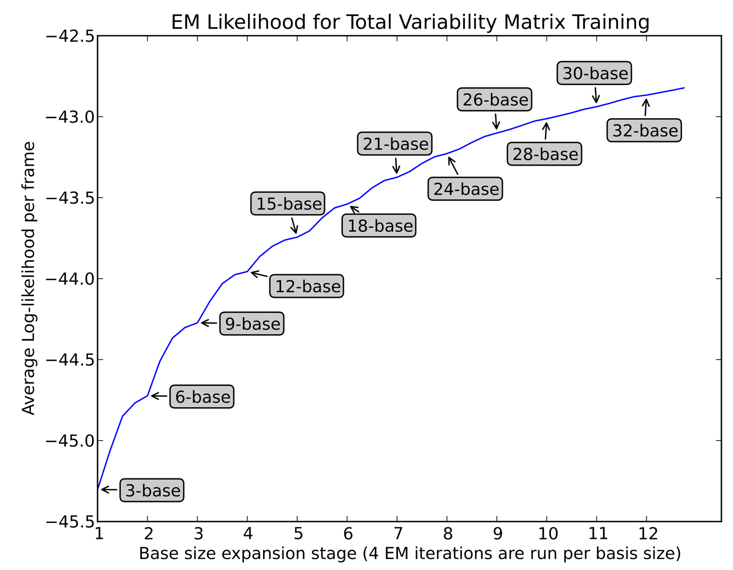
- EM bases estimation solves in the M-step:

$$\sum_{i \in \mathcal{O}} N^i V \mathrm{E}\left[y(i)y(i)^T\right] = \sum_{i \in \mathcal{O}} S^i \mathrm{E}\left[y(i)\right]$$

- Perturb bases in stages, find new orthogonal directions and EM train.

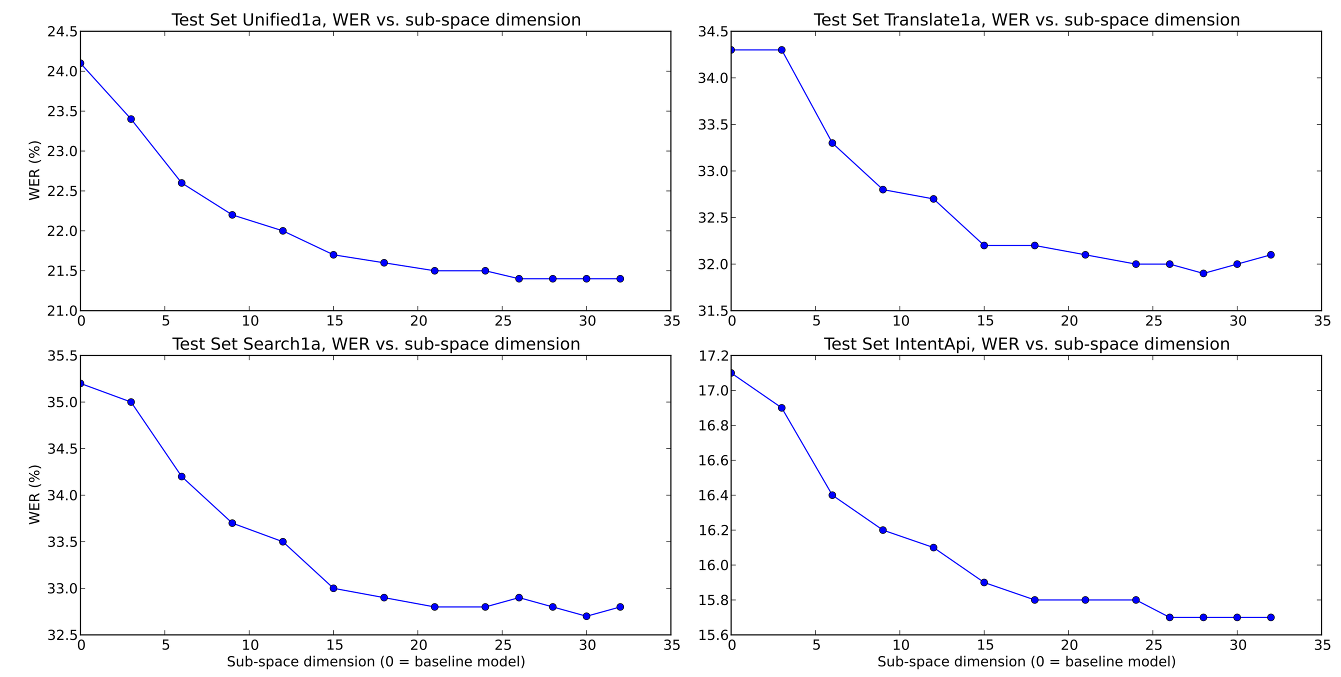


- About 2000 hours of recordings of voice-search, translation and voice-based input method.
- 39-dimensional LDA+STC feature space.
- 2284 tied-state triphone model with 38121 components.
- 32-component model is 290MB



Experiments

Experimental results for four test set, each containing about 20 hours of speech. Per-utterance adaptation (about 4 seconds in duration), no data pooling.



Conclusions

- iVector based adaptation is effective, larger gain than pooled data linear transform-based adaptation.
- Alleviates the need for data pooling which would make the application more complex.
- Large statistics due to the large bases (GMMs) and large fragmentation (per-utterance) can be handled effectively using the Map-Reduce framework.