Adaptive Training Schemes for Robust ASR

Mark Gales

mjfg@eng.cam.ac.uk

CAMBRIDGE UNIVERSITY
ENGINEERING DEPARTMENT

Robust ASR

The *traditional* approach to model-based robust ASR is:

- train a system on clean training data;
- using domain specific data:
  - estimate the acoustic environment parameters;
  - estimate a generic (linear) transform to the environment;
- transform model to new environment.

However we now commonly use found data:

- no control over training data acoustic environment;
- multiple acoustic environments possible;
- multiple speaking styles/speakers.

Acoustic Model Training

Training of acoustic models may be split into two broad groups

- **Multi-Style** training: all the data is treated as a single block. This is the traditional form of training and has proved to be highly successful.
- **Adaptive** training: the data is normalised in some fashion and the system built. Simple normalisation schemes have been used for many years.

Adaptive training is used in all state-of-the-art LVCSR system

The most common form is:

- a simple transformation of the features;
- estimated independently of the model parameters;

This paper examines the use of adaptive training for *Robust ASR* - in particular

Found data for training and testing.

Adaptive Training

Forms of adaptive training are:

- Model independent: no explicit use of any model information - CMN, Variance Normalisation (common, simple form);
- ML Feature transformation: VTLN, constrained MLLR;
- ML Model transformation: the model parameters, means and possibly variances, are transformed - speaker adaptive training (SAT), and cluster adaptive training (CAT).

We can describe the transform *complexities* as:

- simple for the model-independent scheme;
- complex for the feature and model-based schemes.

This talk will concentrate on the issues associated with complex adaptive training schemes.

Forms of Adaptive Training

Scheme	Examples	Hypothesis	Criterion	Training	MAP
Model Independent	CMN, CVN RASTA	$\times$	$\times$	$\surd$	$\times$
Feature-Based	VTLN CMLLR	$\surd$ $(\times)$ $\surd$	$\surd$ $(\times)$ $\surd$	$\surd$	$\times$
Model-Based	MLLR CAT	$\surd$	$\surd$	$\times$	$\surd$

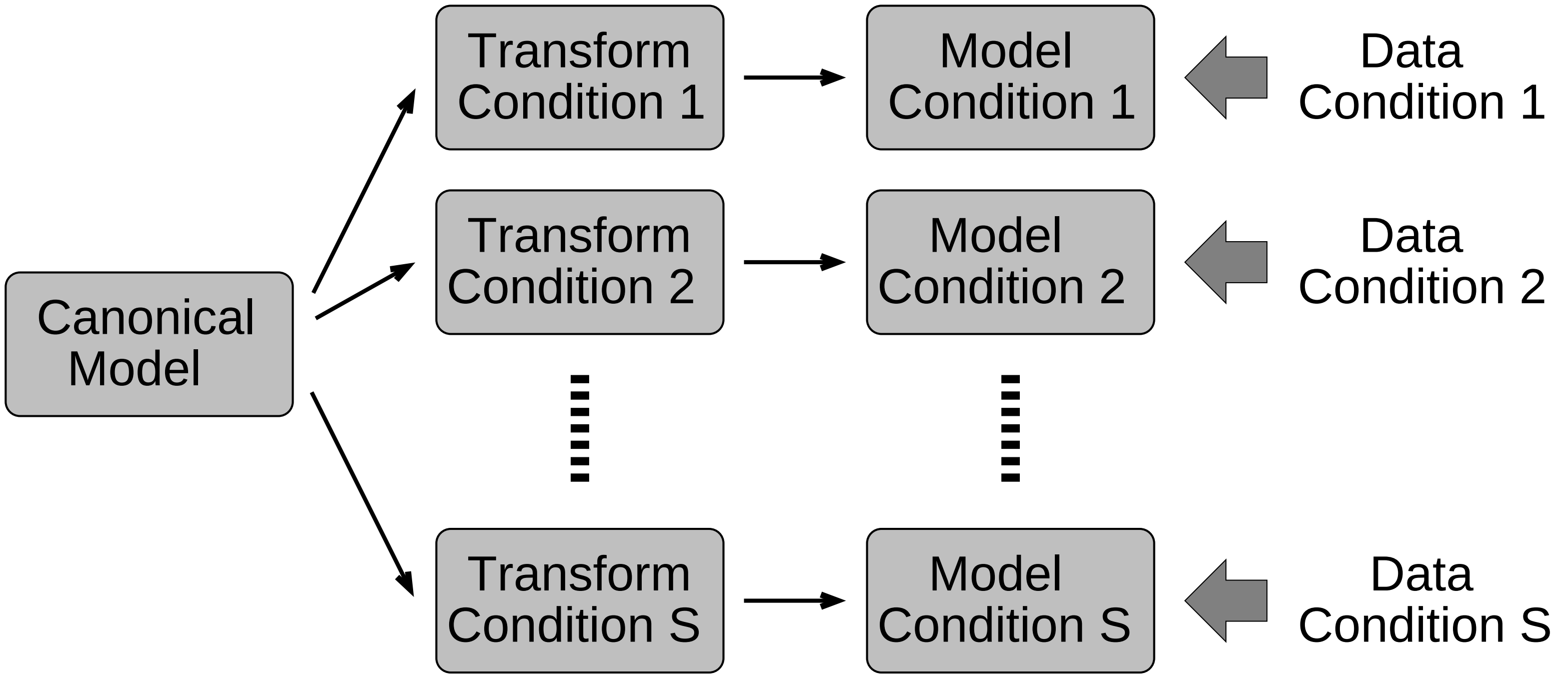
On many tasks the use of more complex adaptive training has been useful.

Why don't we all use "complex" adaptive training?

Two possible reasons

1. It's too hard to implement.
2. It doesn't work for all the conditions we're interested in.

Adaptive Training

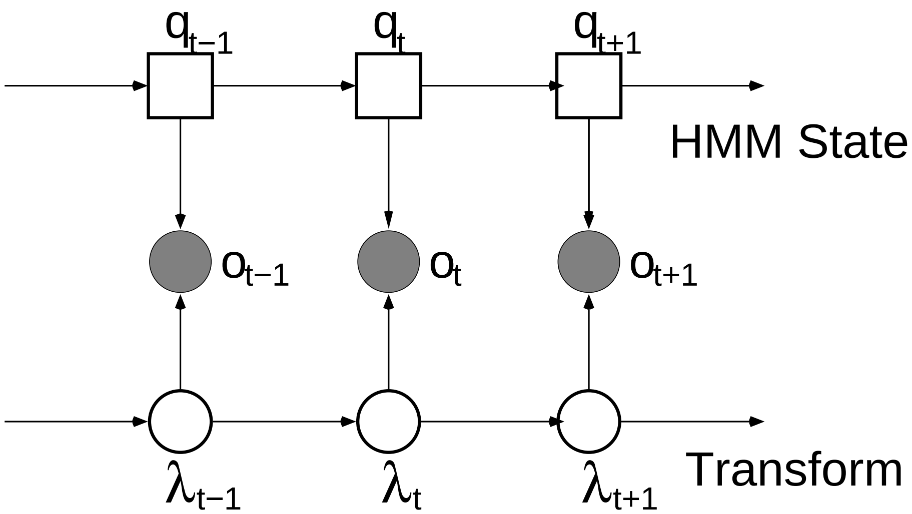


Two sets of "parameters" are extracted from the training data:

- a canonical model is a standard HMM model set;
- a set of transformations, one per condition.

Dynamic Bayesian Network

Alternatively represent as a DBN:



The basic assumption is:

- observations are conditionally independent given
  - the state at that time instance;
  - the transform at that time instance.

Likelihood Calculation

The standard assumption in adaptive training is

- the condition is constant for a complete block of data.

Under this assumption

$$p(\mathbf{O}^{(s)}|\mathcal{M},\tilde{\mathcal{H}}) = \int \sum_{\{\mathbf{Q}^{(\tilde{\mathcal{H}})}\}} \left(\prod_{t=1}^{T} p(\mathbf{o}_t^{(s)}|\boldsymbol{\theta}_{q_t},\boldsymbol{\lambda}) \right) P(\mathbf{Q}|\boldsymbol{\theta}_d) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda}$$

where

- $\boldsymbol{\theta}_d$ are the canonical model parameters for the state durations;
- $\boldsymbol{\theta}_q$ are the PDF parameters associated with state q;
- $\boldsymbol{\phi}$ are the parameters describing the distribution of the transform parameters.

Need to Estimate $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N, \boldsymbol{\theta}_d\}$ and $\boldsymbol{\phi}$

parametric forms are assumed known.

Estimating Model Parameters

Expectation-maximisation (EM) may be used.

Considering only the transform

$$\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) = \sum_{s=1}^{S}\int p(\boldsymbol{\lambda}|\mathbf{O}^{(s)},\mathcal{M},\tilde{\mathcal{H}})\log\left(p(\boldsymbol{\lambda},\mathbf{O}^{(s)}|\hat{\mathcal{M}},\tilde{\mathcal{H}})\right)d\boldsymbol{\lambda}$$

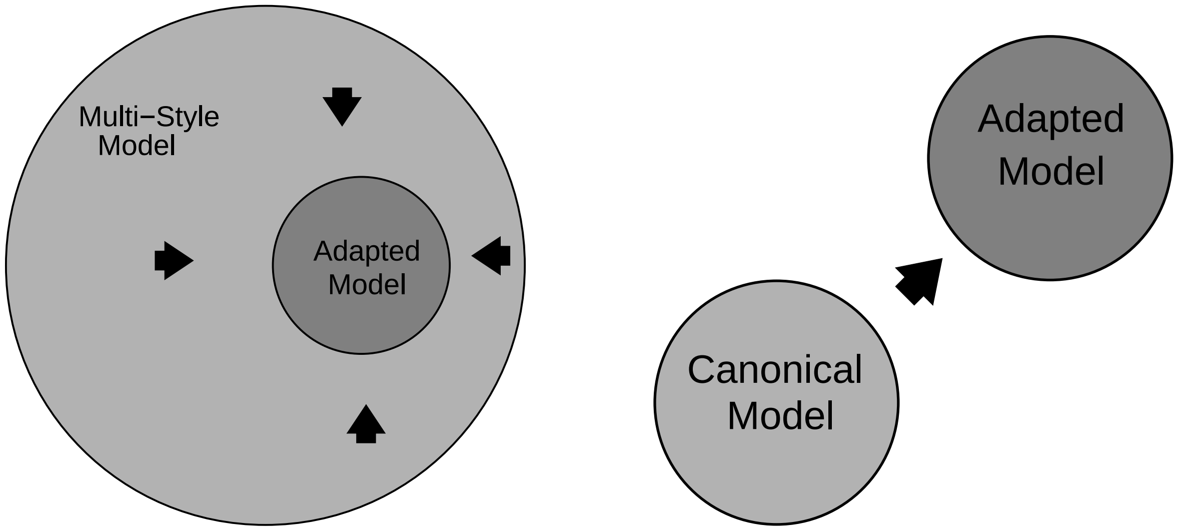
Assume enough training data so

$$p(\boldsymbol{\lambda}|\mathbf{O}^{(s)},\mathcal{M},\tilde{\mathcal{H}}) \approx \delta(\boldsymbol{\lambda}-\boldsymbol{\lambda}_{\text{ML}}^{(s)})$$

We can now write

$$\begin{aligned}\mathcal{Q}(\mathcal{M},\hat{\mathcal{M}}) &= \sum_{s=1}^{S}\log(p(\boldsymbol{\lambda}_{\text{ML}}^{(s)},\mathbf{O}^{(s)}|\hat{\mathcal{M}},\tilde{\mathcal{H}})\\ &= \left(\sum_{s=1}^{S}\log(p(\mathbf{O}^{(s)}|\boldsymbol{\lambda}_{\text{ML}}^{(s)},\hat{\boldsymbol{\Theta}},\tilde{\mathcal{H}})\right)+\left(\sum_{s=1}^{S}\log(p(\boldsymbol{\lambda}_{\text{ML}}^{(s)}|\hat{\boldsymbol{\phi}},\tilde{\mathcal{H}}))\right)\end{aligned}$$

Recognition Stage



(a) Multi-Style System (b) Adaptive System

The requirements of the adaptation transform are very different:

- Multi-Style system: the transform converts a general system to a specific condition;
- Adaptive system: the transform converts a canonical condition to a specific one.

Found Data

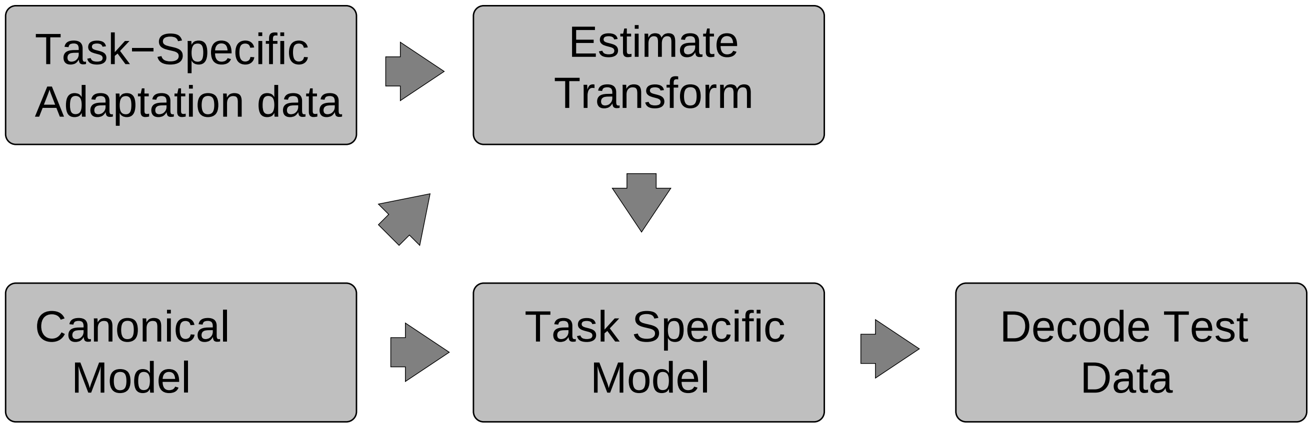
We might expect from *found* data:

- a larger variability in the training data conditions. Thus:
  - the "spread" of the multi-style system increases dramatically;
  - the distance between the canonical model and any specific condition increases.
- in both cases the transform must be more *powerful*, but
  - multi-style system uses the transform to tune to correct condition;
  - adaptive system requires the transform to shift to correct condition.
- adaptive systems are more *sensitive* to the estimate of the transform.

As condition variability increases -

- multi-style system inherently less discriminatory
- adaptive system more sensitive to transform estimate

Adaptation to Specific Task



Two basic forms of adaptation:

- supervised: transcription of the adaptation data known;
- unsupervised: transcription of the adaptation data not known;

Recognition may be run in two modes:

- causal: able to run over the test data only once in order;
- transcription: able to run over the data many times;

Supervised Adaptation

Have some adaptation data $\mathbf{O}^{(a)}$:

- we need the posterior probability of the transform parameters

$$p(\boldsymbol{\lambda}|\mathbf{O}^{(a)}, \mathcal{M}, \mathcal{H})$$

- in practice this may be hard to obtain (linear transforms with complete datasets possible) so
  - MAP Estimates

$$\boldsymbol{\lambda}^{(\mathcal{H})}_{\text{MAP}} = \arg\max_{\boldsymbol{\lambda}} \left(p(\mathbf{O}^{(a)}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) p(\boldsymbol{\lambda}|\phi) \right)$$

  - ML Estimates

$$\boldsymbol{\lambda}^{(\mathcal{H})}_{\text{ML}} = \arg\max_{\boldsymbol{\lambda}} \left(p(\mathbf{O}^{(a)}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) \right)$$

For *found* data supervised adaptation usually not possible.

Unsupervised Adaptation

Unsupervised transcription mode adaptation uses the following inequalities

$$\begin{aligned} p(\mathbf{O}|\mathcal{M}) &= \sum_{\mathcal{H}} \int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda} P(\mathcal{H}) \\ &\leq \int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \check{\mathcal{H}}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda} \\ &\leq p(\mathbf{O}|\boldsymbol{\lambda}_{\mathrm{MAP}}^{(\check{\mathcal{H}})}, \boldsymbol{\Theta}, \check{\mathcal{H}}) \\ &\leq p(\mathbf{O}|\boldsymbol{\lambda}_{\mathrm{ML}}^{(\check{\mathcal{H}})}, \boldsymbol{\Theta}, \check{\mathcal{H}}) \end{aligned}$$

where

$$\check{\mathcal{H}} = \arg\max_{\mathcal{H}} \left(\int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda} P(\mathcal{H}) \right)$$

Decoding is then performed using

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left(p(\mathbf{O}|\boldsymbol{\lambda}_{\mathrm{ML}}^{(\check{\mathcal{H}})}, \boldsymbol{\Theta}, \mathcal{H}) P(\mathcal{H}) \right)$$

Unsupervised Adaptation (cont)

So what's the problem?

- usually cannot find $\check{\mathcal{H}}$, approximation must be used, normally

$$\check{\mathcal{H}} = \arg\max_{\mathcal{H}} \left(p(\mathbf{O}|\boldsymbol{\Theta}_{\text{Mult}}, \mathcal{H})P(\mathcal{H})\right)$$

  where $\boldsymbol{\Theta}_{\text{Mult}}$ are the multi-style model parameters.

- we can only ensure that

$$p(\mathbf{O}|\boldsymbol{\lambda}_{\text{ML}}^{(\check{\mathcal{H}})}, \boldsymbol{\Theta}, \hat{\mathcal{H}})P(\hat{\mathcal{H}}) \geq p(\mathbf{O}|\boldsymbol{\lambda}_{\text{ML}}^{(\check{\mathcal{H}})}, \boldsymbol{\Theta}, \check{\mathcal{H}})P(\check{\mathcal{H}})$$

- If the transform is too complex
  - or estimated on too little data

  then the hypothesis won't change

$$\hat{\mathcal{H}} = \check{\mathcal{H}}$$

We've got nowhere!

Relaxing the Approximations

How do we handle this problem;

- Use schemes from unsupervised adaptation .....
- in terms of the inequalities

$$\begin{aligned} p(\mathbf{O}|\mathcal{M}) &= \sum_{\mathcal{H}} \int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) p(\boldsymbol{\lambda}|\phi) d\boldsymbol{\lambda} P(\mathcal{H}) && \text{Lattice-based adaptation} \\ &\leq \int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \check{\mathcal{H}}) p(\boldsymbol{\lambda}|\phi) d\boldsymbol{\lambda} && \text{Posterior adaptation} \\ &\leq p(\mathbf{O}|\boldsymbol{\lambda}^{(\check{\mathcal{H}})}_{\text{MAP}}, \boldsymbol{\Theta}, \check{\mathcal{H}}) && \text{MAP adaptation} \\ &\leq p(\mathbf{O}|\boldsymbol{\lambda}^{(\check{\mathcal{H}})}_{\text{ML}}, \boldsymbol{\Theta}, \check{\mathcal{H}}) && \text{ML adaptation} \end{aligned}$$

- lattice-based adaptation exclusive to unsupervised adaptation;
- posterior, MAP and ML suited for supervised adaptation as well.

Consistent framework for techniques

Posterior Adaptation

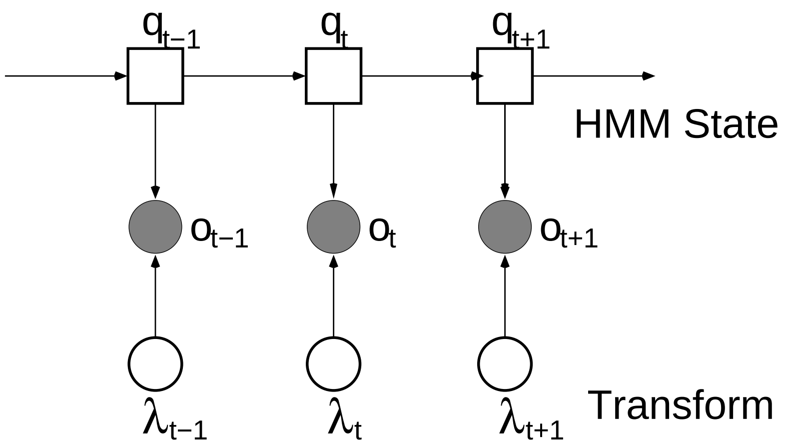
Rather than taking the ML or MAP estimate of the transform parameters use the *posterior distribution* over the transform parameters.

Decode using

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left(\int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) p(\boldsymbol{\lambda}|\mathcal{M}, \check{\mathcal{H}}, \mathbf{O}) d\boldsymbol{\lambda} P(\mathcal{H}) \right)$$

- related to Bayesian prediction adaptation;
- handles the initial decode problem;
- unfortunately no simple solution!

Posterior Adaptation (cont)



- simplifies system by assuming independence;
- similar to standard HMM training (with a few assumptions!);

$$p(\mathbf{O}|\mathcal{M},\mathcal{H}) \approx \sum_{\{\mathbf{Q}^{(\mathcal{H})}\}} \left(\prod_{t=1}^{T} \check{p}(\mathbf{o}_t|\boldsymbol{\theta}_{q_t}) \right) P(\mathbf{Q}|\boldsymbol{\theta}_d)$$

$\check{\boldsymbol{\mu}}^{(qmp)} = \mathcal{E}\left\{\mathbf{o}|q,m,p\right\}$ etc.

Discriminative Adaptive Training

MMI training has been shown to outperform ML training

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \left(\sum_{s=1}^{S} \log \left(\frac{p(\mathbf{O}^{(s)}|\mathcal{M}, \tilde{\mathcal{H}})P(\tilde{\mathcal{H}}))}{\sum_{\mathcal{H}} p(\mathbf{O}^{(s)}|\mathcal{M}, \mathcal{H})P(\mathcal{H})} \right) \right)$$

Can be used in an adaptive training

- estimate transform in a discriminative fashion;
- estimate canonical model in a discriminative fashion;

Problems with unsupervised adaptation

How to discriminatively train the target transform?

Alternative scheme

- estimate transform in an ML fashion;
- estimate canonical model in a discriminative fashion;

Future (Present?)

- Adaptive training will become more important as greater use is made of *found* data.
- Using adaptive training for robust ASR places more emphasis on the transform.
- Unsupervised schemes and extensions can reduce this problem.
- Current transforms are inadequate - the world is not linear.
- Discriminative training consistently better than ML, more extensive use of *discriminative adaptive* training.
- Schemes for overcoming sensitive of adaptively trained schemes essential.
- More complex dynamic Bayesian networks for time-varying acoustic conditions.

CoreTex

CoreTex -

Improving Core Speech Recognition Technology

Main areas:

- Genericity and Adaptability: build systems that work well on a wide range of tasks and dynamic adaptation;
- Portability: porting systems to a new domain with "reasonable" cost;
- Enriched Transcription: more information than simple textual transcription.

Web-Site: http://coretex.itc.it

Participants

- Rheinisch-Westfaelische Technische Hochshule Aachen - RWTH
- University of Cambridge - CUED
- Istituto Trentino di Cultura - IRST
- Centre National de la Recherche Scientifique CNRS/LIMSI: