Constrained Estimation of Hidden Markov Models

Mark Gales

mjfg@watson.ibm.com

T.J. Watson Research Center

Talk Layout

1. **"Traditional" constraints** - tying, diagonal covariance matrices, speaker clustering.
2. **Covariance matrix modelling** - semi-tied covariance matrices, factor analysis, shared factor analysis.
3. **Speaker adaptation** - MLLR, general linear transforms.
4. **Adaptive training** - SAT, CAT.

Talk assumes Gaussian-component continuous-density HMMs

Traditional constraints

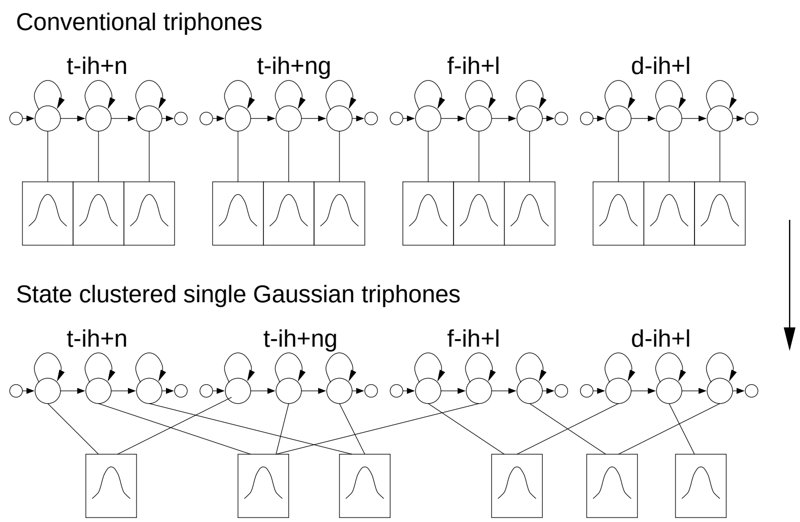
These tend to be

Hard - Parameters are identical, independent or zero.

Examples are

1. **State clustering** - states are tied together according to their phonetic context.
2. **Speaker clustering** - speakers are assigned to one of a set of speaker groups.
3. **Diagonal covariance matrices** - set all off-diagonal elements to zero.

State clustered triphones



1. Use a **decision tree** to determine which triphones are clustered together.
2. **Merge** all statistics from each set of clustered states to estimate a single "tied" state.

Good for large amounts of data

How to handle limited data?

Nature of the Constraints

Would like contraints to:

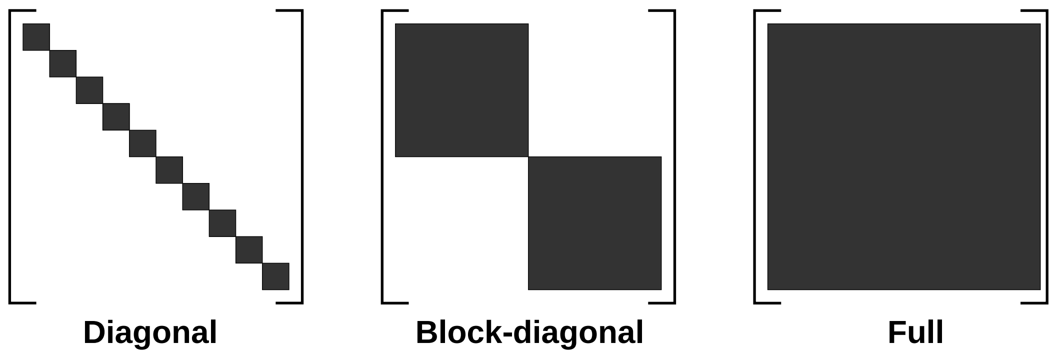
- be related to whatever is being modelled.
- result in as few parameters as possible.
- have simple ML estimation formulae.
- be efficient at run-time decoding.

Two examples considered **Covariance Modelling** and **Speaker Adaptation**.

Covariance Modelling

Standard choice

1. **Diagonal** - assumes each element is independent.
2. **Block-diagonal** - assumes elements may be grouped, groups are independent.
3. **Full** - explicitly models all correlation in the feature vector.



Covariance Modelling

Diagonal	Few parameters n
	Likelihood calculation $\mathcal{O}(n)$.
Block-Diagonal	Parameters $\frac{n^2+kn}{2k}$
	Likelihood calculation $\mathcal{O}(\frac{n^2}{k})$.
Full	Parameters $\frac{n^2+n}{2}$
	Likelihood calculation $\mathcal{O}(n^2)$.

Given a fixed amount of training data:

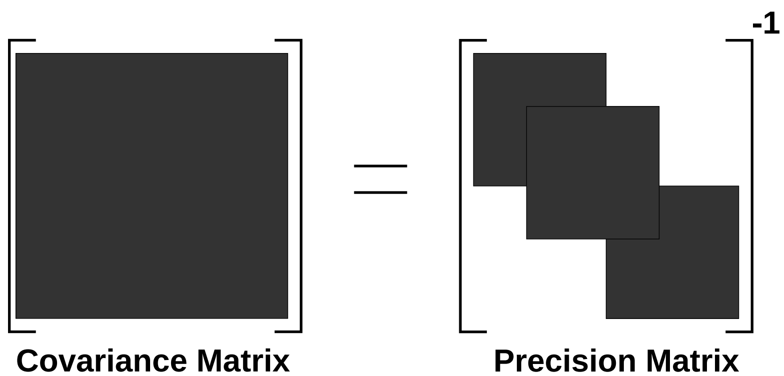
Balance Complexity vs. Number Distributions

Usually elements are correlated - other choices?

Precision Matrices

Models data of the form

$$\begin{aligned} x_1 &= \mathcal{N}(\mu_1, \sigma_1^2) \\ x_2 &= x_1 + \mathcal{N}(\mu_2, \sigma_2^2) \\ x_3 &= x_2 + \mathcal{N}(\mu_3, \sigma_3^2) \end{aligned}$$



Decode using

$$\mathcal{L}(\mathbf{o}|m) = \frac{\prod_{\mathcal{C}} \mathcal{L}^{\mathcal{C}}(\mathbf{o}^{\mathcal{C}})}{\prod_{\mathcal{S}} \mathcal{L}^{\mathcal{S}}(\mathbf{o}^{\mathcal{C}})}$$

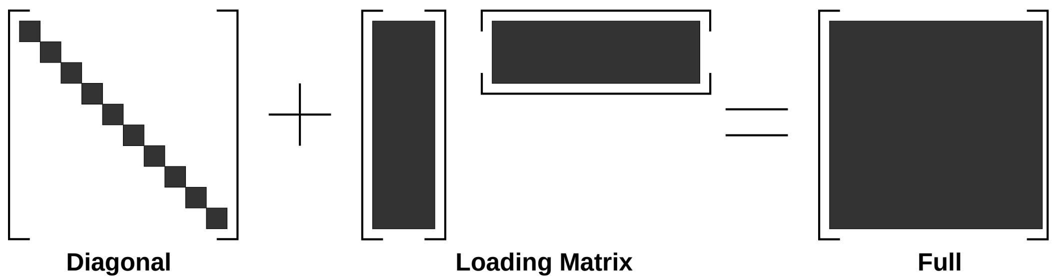
No simple way to find structure?

Factor analysis

Structure

$$\boldsymbol{\Sigma}^{(m)} = \boldsymbol{\Sigma}_{\text{diag}}^{(m)} + \boldsymbol{\Lambda}^{(m)}\boldsymbol{\Lambda}^{(m)T}$$

Loading matrix is only $n \times k$



Decode cost

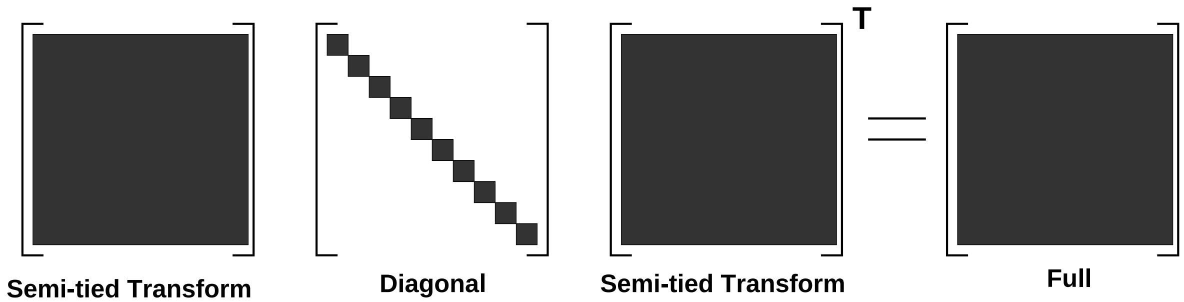
Component n-dimensional diagonal **plus** k-dimensional full computation

Component $n \times k$ matrix-vector multiplication

Semi-tied Covariance matrices

Has the form

$$\boldsymbol{\Sigma}^{(m)} = \mathbf{H}^{(r)}\boldsymbol{\Sigma}^{(m)}_{\text{diag}}\mathbf{H}^{(r)}$$



Decode using

$$\mathcal{L}(\mathbf{o}|m) = \frac{1}{|\mathbf{H}^{(r)}|}\mathcal{N}(\mathbf{H}^{(r)-1}\mathbf{o}; \mathbf{H}^{(r)-1}\mu^{(m)}, \boldsymbol{\Sigma}^{(m)}_{\text{diag}})$$

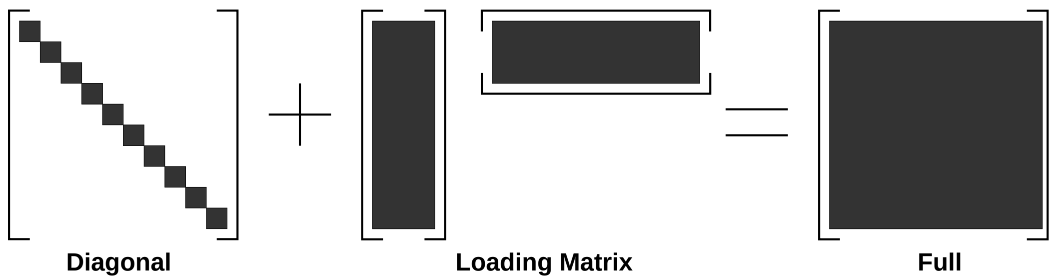
Component	n-dimensional diagonal calculation
Transform	$n \times n$ matrix-vector multiplication

Shared Factor Analysis

Structure

$$\mathbf{\Sigma}^{(m)} = \mathbf{\Sigma}^{(m)}_{\text{diag}} + \mathbf{\Lambda}^{(r)}\mathbf{\Lambda}^{(r)T}$$

where **Loading Matrices** are shared at say a state level.

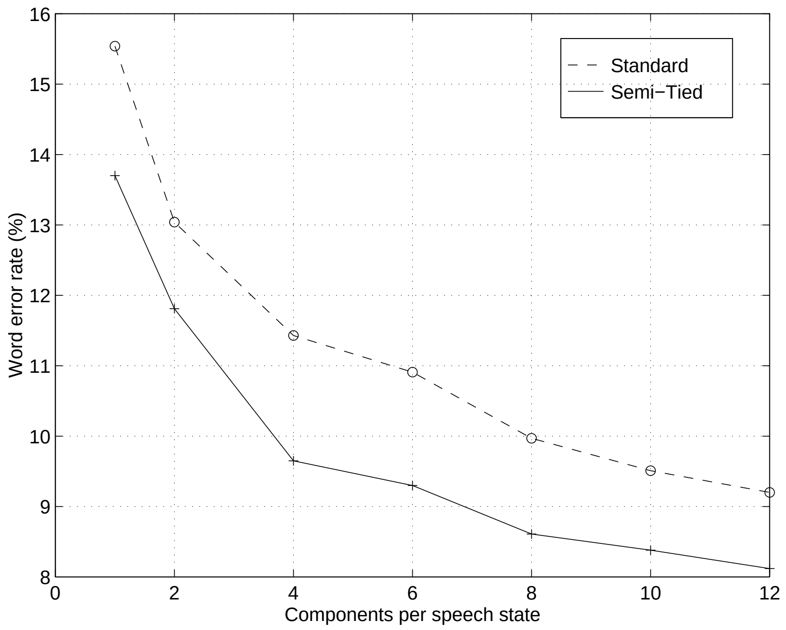


Decode cost

Component n-dimensional diagonal **plus** k-dimensional full computation

Component $n \times k$ matrix vector multiplication

Semi-Tied Covariance Results - WSJ94



1. Same performance with approx. half the number of parameters.

2. With approx. same parameters 10% reduction in word error rate.

Covariance Modelling

What is the nature of the data?

What is the best use of the data?

What is the run-time computational load?

Speaker Adaptation

Speaker-Dependent systems outperform **Speaker Independent** systems.

However:

Usually insufficient data to robustly estimate speaker dependent models

Solution:

Speaker Adaptation - use “constraints” to reduce number of free parameters

(Some schemes also applicable to **Environment Adaptation**)

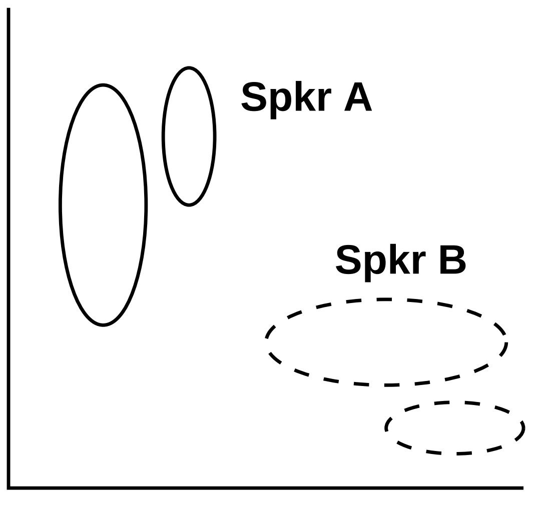
Examples

A number of constraints have been examined

1. **Vocal tract normalisation** - differences between speakers may be represented as a frequency axis warping (possibly non-linear).
2. **Extended MAP** - correlations between parameters of all speakers may be represented as a prior distribution.
3. **Speaker clustering** - speaker is assumed to be a member of a set of disjoint speaker groups.
4. **Linear transforms**- differences between speakers is a linear (or piecewise-linear) transformation of the model parameters.

Linear Transform Constraints

A linear transform of a **Canonical** model = **Speaker dependent** model.



- Simple to estimate ML transform parameters using EM.
- It works for small amounts of adaptation data.

Usually **Speaker Independent** models used as the canonical models.

Linear Transforms

Choices:

1. **Structure of transform**:
   Select structure - diagonal, block-diagonal, full, shifting mel-bin center frequencies, etc.

2. **Grouping distributions**:
   Which distributions have the same transform - expert knowledge, position in acoustic space, hard/soft groupings, ML groupings, etc.

3. **Number of transforms**:
   Given some adaptation data how many transforms - global, fixed number, regression class tree, observation weightings, etc.

Balance Complexity vs. Number Transforms

Maximum Likelihood Linear Regression

The differences between speakers is a piecewise-linear transform of the means -

$$\hat{\mu}^{(m)} = \mathbf{A}^{(r)}\mu^{(m)} + \mathbf{b}^{(r)}$$

Decode using

$$\mathcal{L}(\mathbf{o}|m) = \mathcal{N}(\mathbf{o}; \hat{\mu}^{(m)}, \mathbf{\Sigma}^{(m)})$$

Simple to estimate, **but** need to modify models.

Variance Adaptation

Variance adaptation may be performed in addition to MLLR.

Iterative optimisation

1. Estimate **MLLR transform** given current variance transform.
2. Estimate **Variance transform** given current MLLR transform.

Typically the variances are compressed (eg in additive noise, or speaker adaptation), so applying only a bias is not usually appropriate.

Normalised-Full Variance Adaptation

Here ($\mathbf{L}^{(m)}$ is the Choleski factorisation, $\mathbf{H}^{(r)}$ is the transform)

$$\hat{\boldsymbol{\Sigma}}^{(m)} = \mathbf{L}^{(m)}\mathbf{H}^{(r)}\mathbf{L}^{(m)T}$$

Simple to estimate, however **decode** using

$$\mathcal{L}(\mathbf{o}|m) = \mathcal{N}(\mathbf{o}; \hat{\mu}^{(m)}, \hat{\boldsymbol{\Sigma}}^{(m)})$$

Full transform = Full matrix calculation

BUT without memory requirements

Efficient-Full Variance Transform

Has the form

$$\hat{\mathbf{\Sigma}}^{(m)} = \mathbf{H}^{(r)}\mathbf{\Sigma}^{(m)}\mathbf{H}^{(r)T}$$

Harder to estimate, however **decode** using

$$\mathcal{L}(\mathbf{o}|m) = \frac{1}{|\mathbf{H}^{(r)}|}\mathcal{N}(\mathbf{H}^{(r)-1}\mathbf{o}; \mathbf{H}^{(r)-1}\hat{\mu}^{(m)}, \mathbf{\Sigma}^{(m)})$$

Full transform = Diagonal matrix calculation

BUT need to transform observations

When the transforms are diagonal, this is the same as the normalised-full variance transform.

Constrained Model-Space Transform

Also known as a **Feature-Space** transform.

Mean and variance have the same linear transform -

$$\begin{aligned} \hat{\mu}^{(m)} &= \mathbf{A}^{(r)}\mu^{(m)} + \mathbf{b}^{(r)} \\ \hat{\mathbf{\Sigma}}^{(m)} &= \mathbf{A}^{(r)}\mathbf{\Sigma}^{(m)}\mathbf{A}^{(r)T} \end{aligned}$$

Harder to estimate, however **decode** using

$$\mathcal{L}(\mathbf{o}|m) = \frac{1}{|\mathbf{A}^{(r)}|}\mathcal{N}(\mathbf{A}^{(r)-1}(\mathbf{o} - \mathbf{b}^{(r)}); \mu^{(m)}, \mathbf{\Sigma}^{(m)})$$

No need to modify the models

Adaptation Results

Transform Set	Form	Error Rate (%)		
		H1 Dev	H1 Eval	S5 Eval
—	—	9.57	9.20	8.95
Feature	Diagonal	8.47	8.48	7.99
	Block	8.14	7.75	7.62
MLLR	Diagonal	8.61	8.48	7.93
	Block	8.06	8.13	7.15

Incremental Adaptation

Speaker Adaptation

How much adaptation data?

What is the adaptation mode?

What is the run-time computational load?

What is the nature of the data (environment)?

Adaptive training

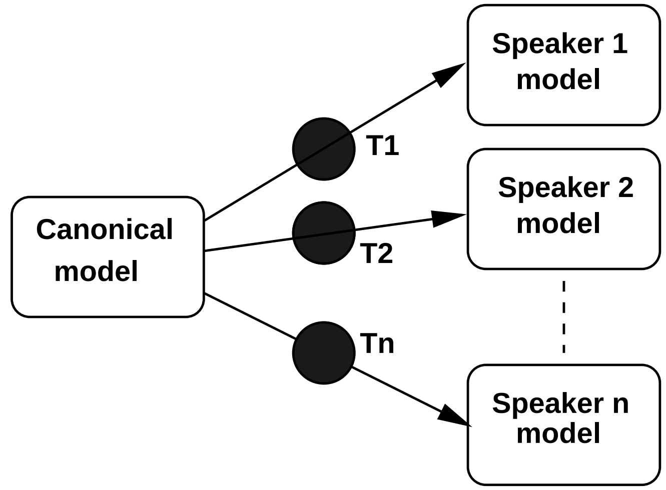
For non-homogeneous training data (e.g. multiple training speakers)

Use constraints in training

Should

- allow only intra-speaker variability to be modelled.
- allow fewer parameters.
- reduce word error rate.

Adaptive Training



An iterative scheme is used:-

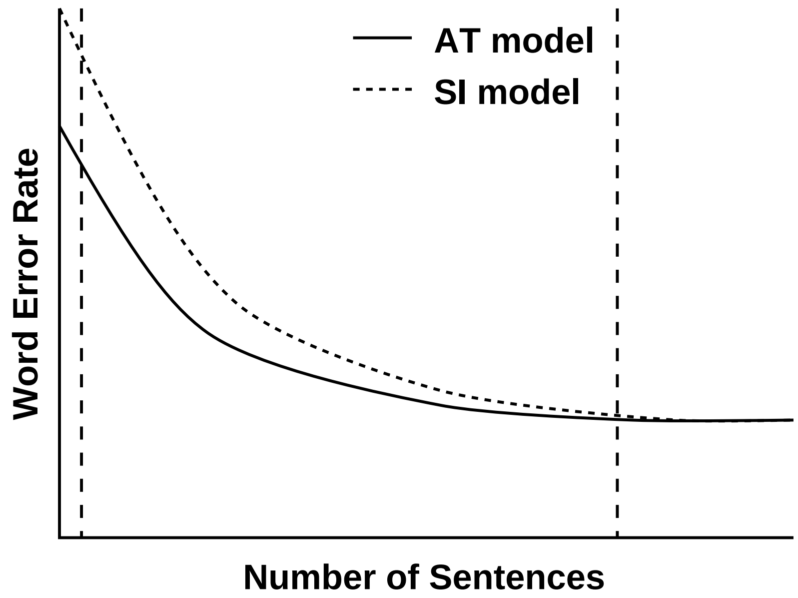
1. Estimate the **transform parameters** for all training speakers.

2. Estimate the **canonical model** given the constraint parameters.

Solutions for the first stage already detailed,

How to estimate the canonical model?

Error rate prediction



Three regions

- **Loads of data** - estimate large number of transforms
- **Some data** - estimate a few transforms
- **Little/no data** - estimate no transforms!

Speaker Adaptive Training

Use MLLR for the mean constraints. Canonical model given by

$$\hat{\mu}^{(m)} = \left(\sum_{s,\tau} \gamma_m(\tau) \mathbf{A}^{(s)T} \hat{\mathbf{\Sigma}}^{(m)-1} \mathbf{A}^{(s)} \right)^{-1} \sum_{s,\tau} \gamma_m(\tau) \mathbf{A}^{(s)T} \hat{\mathbf{\Sigma}}^{(m)-1} \left(\mathbf{o}(\tau) - \mathbf{b}^{(s)} \right)$$

and

$$\hat{\mathbf{\Sigma}}^{(m)} = \frac{\sum_{s,\tau} \gamma_m(\tau)(\mathbf{o}(\tau) - \hat{\mu}^{(sm)})(\mathbf{o}(\tau) - \hat{\mu}^{(sm)})^T}{\sum_{s,\tau} \gamma_m(\tau)}$$

where $\hat{\mu}^{(sm)} = \mathbf{A}^{(s)} \hat{\mu}^{(m)} + \mathbf{b}^{(s)}$

Problem

Sufficient statistics required at a speaker level

Modified SAT

Use **feature-space** transform for the constraints

$$\hat{\mu}^{(m)} = \frac{\sum_{s,\tau} \gamma_m(\tau)\hat{\mathbf{o}}^{(s)}(\tau)}{\sum_{s,\tau} \gamma_m(\tau)}$$

and

$$\hat{\mathbf{\Sigma}}^{(m)} = \frac{\sum_{s,\tau} \gamma_m(\tau)(\hat{\mathbf{o}}^{(s)}(\tau) - \hat{\mu}^{(m)})(\hat{\mathbf{o}}^{(s)}(\tau) - \hat{\mu}^{(m)})^T}{\sum_{s,\tau} \gamma_m(\tau)}$$

where $\hat{\mathbf{o}}^{(s)}(\tau) = \mathbf{A}^{(s)}\mathbf{o}(\tau) + \mathbf{b}^{(s)}$

Only standard HMM training statistics required

Cluster Adaptive Training

Simplify transformation of the mean to

$$\hat{\mu}^{(m)} = \left[\begin{array}{ccc} \mu^{(m1)} & \dots & \mu^{(mP)} \end{array} \right] \lambda^{(r)}$$

where

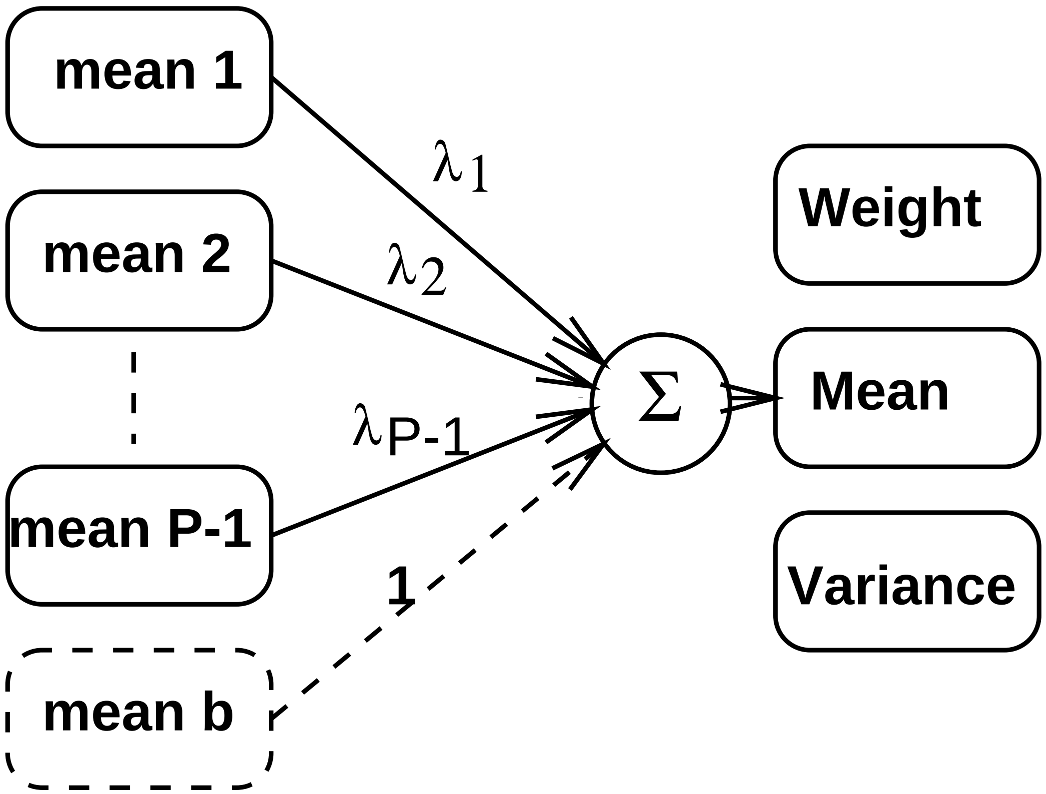
$$\lambda^{(r)} = \left[\begin{array}{cccc} \lambda_1^{(r)} & \dots & \lambda_{P-1}^{(r)} & 1 \end{array} \right]^T$$

May be viewed as **continuous speaker clustering**

Restrictions

- Covariance matrices tied over clusters
- Gaussian component priors tied over clusters

Cluster Adaptive Training



Has very few parameters

Transform-Based Clusters

Use MLLR Transforms to represent clusters

$$\hat{\mu}^{(m)} = \left(\sum_p \lambda_p^{(r)} \mathbf{A}^{(rp)}\right) \mu^{(m)} + \sum_p \lambda_p^{(r)} \mathbf{b}^{(rp)}$$

Related to SAT with

- Fixed number of "speakers" (solves training)
- Smoothing between transforms

Canonical model estimation similar to SAT

SAT Results - WSJ 1994

Canonical Model	Number Transforms	Error Rate (%)	
		H1 Dev	H1 Eval
SI	1	9.07	7.97
	2	8.64	7.73
SAT	1	8.26	7.26
	2	8.00	7.09

Unsupervised Static Adaptation

SAT Results - Hub4 1997

Canonical Model	Adaptation Scheme	Error Rate (%) Hub4 1997
SI	—	20.9
	MLLR+EVAR	18.8
	Feature	19.3
SAT	Feature	18.5
	Feature+MLLR	18.0

Unsupervised Static Adaptation

CAT Results - IBM Internal

Canonical Model	Adaptation Scheme	Error Rate (%) IBM Internal
SI	—	12.6
	Feature+MLLR	10.0
CAT 1+1	—	11.7
	Feature+MLLR	9.4

Supervised Static Adaptation

Handling no Adaptation Data?

Bayesian interpretation of adaptive training

$$\mathcal{L}(\mathbf{O}) = \int \left(\prod_{\tau} \mathcal{L}(\mathbf{o}(\tau)|\lambda) \right) \mathcal{L}(\lambda|\mathcal{M}) d\lambda$$

where λ are the parameters of the constraints used.

With simple assumptions model may be trained, note

- Cost of maintaining and updating parameter posterior distributions
- No closed-form for integration
- Removes need for SI models

Future Work

Adaptive Training

Useful depending on amount of adaptation data

Increases complexity of training

How to handle initial alignment if unsupervised?

Summary

Constrained estimation of HMMs is useful