Linear Transformations (Yet Again!!)

Mark Gales

CAMBRIDGE UNIVERSITY
ENGINEERING DEPARTMENT

mjfg@eng.cam.ac.uk

Introduction

Linear transformations within HMM systems have been shown to be very useful.

- **Adaptation**: MLLR, SAT, CAT, Cluster Transform etc etc.
- **Feature space generation**: semi-tied systems, HLDA etc etc.

This talk will examine yet more variations:

- **Multiple ML projection schemes**
- **Factored semi-tied covariance matrices**
- **Multiple-cluster adaptive training schemes**

All schemes will be trained using ML estimation.

ML Projection Schemes

Linear projection schemes

- **Linear discriminant analysis**
- **Heteroscedastic LDA**
- **Heteroscedastic discriminant analysis** (not ML);

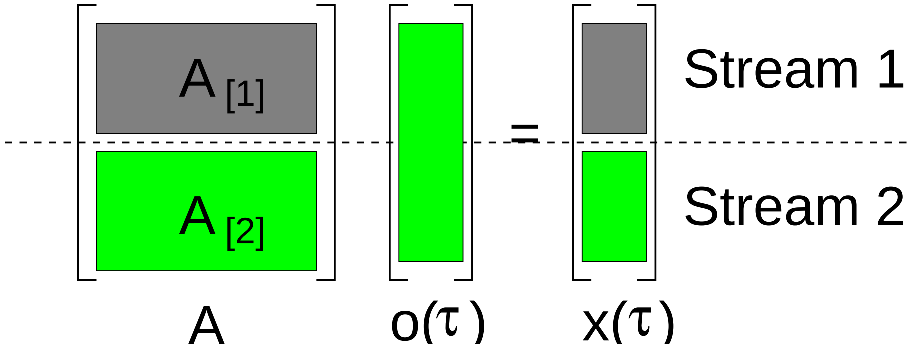
This talk will be concerned with using multiple projections in a consistent ML framework.

Motivation - why use same features for all parts of speech?

Problem - need to be able to compare likelihoods.

Multiple "Streams"

Standard ML projections use 2 streams

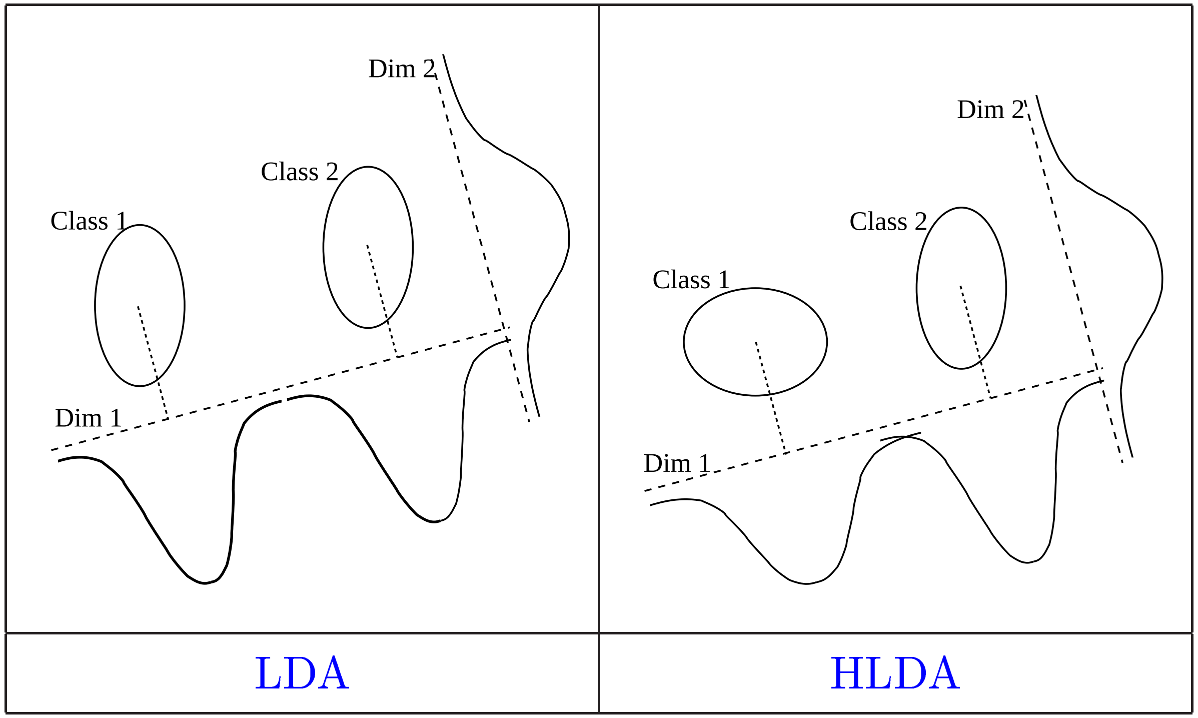


Streams are labelled as:

- **Speech** - stream 1, class dependent models
- **Noise** - stream 2, global models

Standard noise model is a single multivariate Gaussian (though not a restriction).

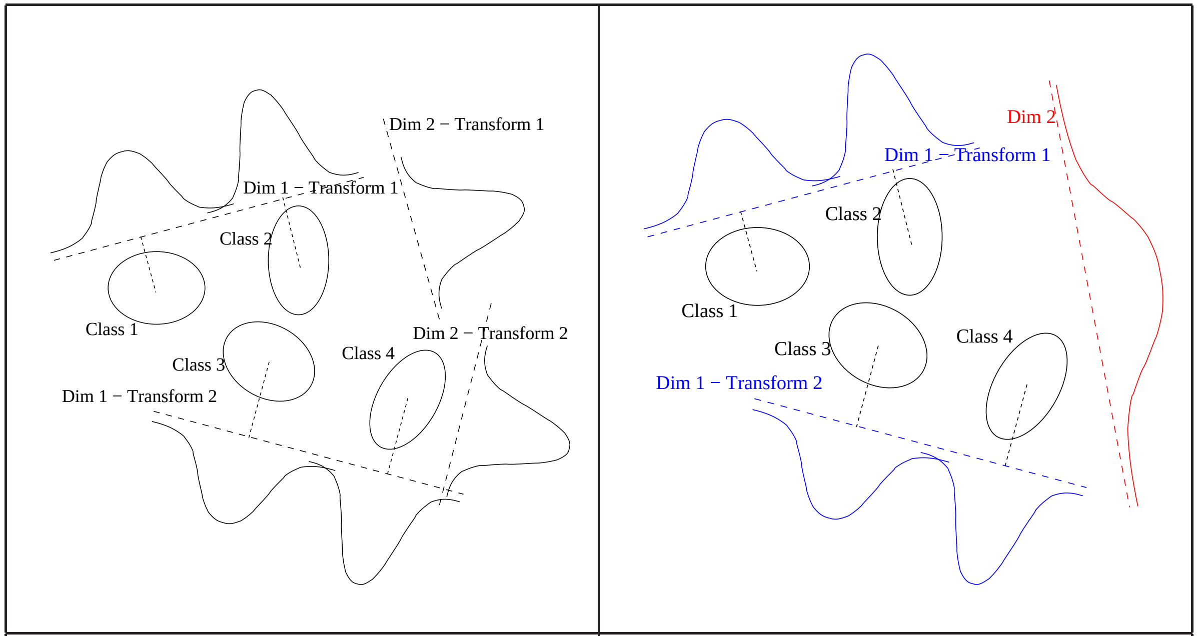
Standard ML Projections



Two standard forms of single projection:

- **LDA**: common class covariance matrices (approximate)
- **Heteroscedastic LDA** (HLDA): class dependent covariances

Multiple ML Projections



Multiple HLDA	Multiple LDA

Two forms of multiple projection considered:

- **Multiple HLDA** (MHLDA): perform HLDA per group
- **Multiple LDA** (MLDA): use a common "noise" stream

Multiple HLDA

Assume:

- there are R component sets
- projecting from n to p dimensions

Compute likelihood as

$$\mathcal{L}(\mathbf{o}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}_{\text{diag}}^{(m)}, \mathbf{A}^{(r)}) = l^{(r)}|\det(\mathbf{A}^{(r)})|\mathcal{N}(\mathbf{A}_{[1]}^{(r)}\mathbf{o}; \boldsymbol{\mu}_{[1]}^{(m)}, \boldsymbol{\Sigma}_{\text{diag}[1]}^{(m)})$$

where

$$l^{(r)} = \mathcal{N}(\mathbf{A}_{[2]}^{(r)}\mathbf{o}; \boldsymbol{\mu}_{[2]}^{(r)}, \boldsymbol{\Sigma}_{\text{diag}[2]}^{(r)})$$

$l^{(r)}$ need only be calculated once for each transform class.

Multiple LDA

Assume:

- there are R component sets
- projecting from n to p dimensions

Compute likelihood as

$$\mathcal{L}(\mathbf{o};\boldsymbol{\mu}^{(m)},\boldsymbol{\Sigma}_{\text{diag}}^{(m)},\mathbf{A}^{(r)}) = |\det(\mathbf{A}^{(r)})|\mathcal{N}(\mathbf{A}_{[1]}^{(r)}\mathbf{o};\boldsymbol{\mu}_{[1]}^{(m)},\boldsymbol{\Sigma}_{\text{diag}[1]}^{(m)})$$

Advantages:

- likelihood computed in p-dimensional space
- reduced number of model parameters
- simple to extend to a hierarchy of transforms

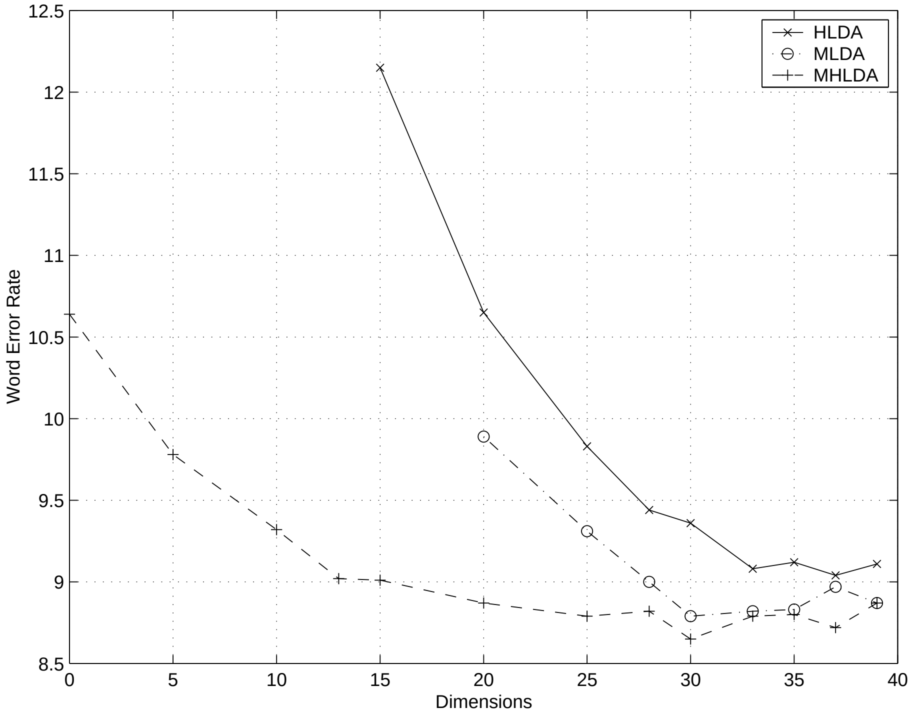
Initial Experiments

Experimental details:

- Standard LVCSR task - Wall Street Journal (1994)
- Cross-word, state-clustered, triphone system
- 6403 distinct states generated
- State-level transform classes ($R = 6403$)
- 12 Gaussian components per state ($\approx$ 75K)
- MFCC + log-energy + delta + delta-delta, CMN (39-dimensional)
- "Expert" transform class assignment

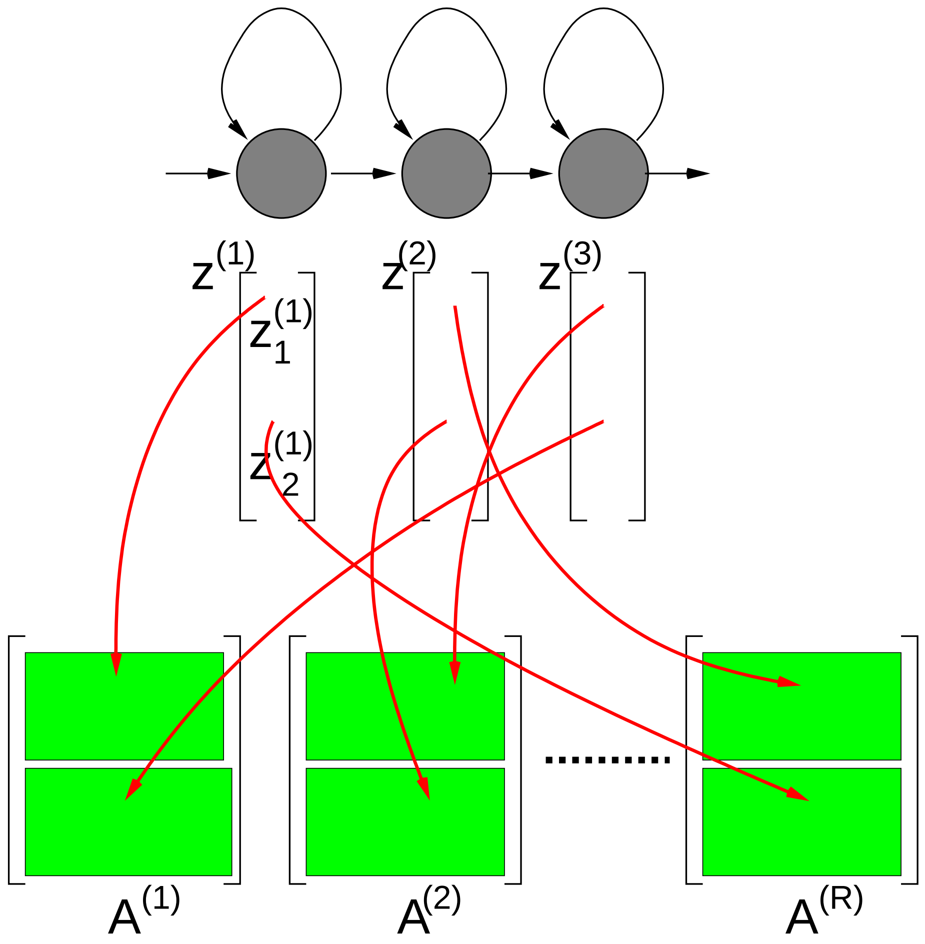
Only single pass retraining currently used (sub-optimal!!)

Results



- Baseline performance 9.71%
- MHLDA limits as a single full covariance system
- No statistical difference (at 95% level) between systems

Factored Semi-Tied Coariance Matrices



Again partition transform - why use both partitions from same transform

Training

An iterative, generalised EM, scheme is used.

1. Initialise alignments using "expert" knowledge and train SI model
2. Given alignments estimate the transform parameters (iterative): more complex than standard semi-tied systems
3. Update the transform alignments given the alignements: not possible to test all R^S transforms - iterative scheme used
4. Given the transforms update the component covariance matrices
5. Goto (2) unless converged

Alignment Estimation

Component assignment described by an indicator *vector*, $\mathbf{z}^{(m)}$.

1. Update assignment for each stream, s, for each component, m,

$$z_s^{(m)} = \arg\max_{r \in R_s} \left\{ \left(\frac{\left|\det\left(\boldsymbol{A}^{(\boldsymbol{u}^{(srm)})}\right)\right|^2}{\left|\det\left(\operatorname{diag}\left(\boldsymbol{A}_{[s]}^{(r)} \boldsymbol{W}^{(m)} \boldsymbol{A}_{[s]}^{(r)T}\right)\right)\right|} \right) \right\}$$

   where

$$u_j^{(srm)} = \begin{cases} r, & j = s \\ z_j^{(m)}, & \text{(otherwise)} \end{cases}$$

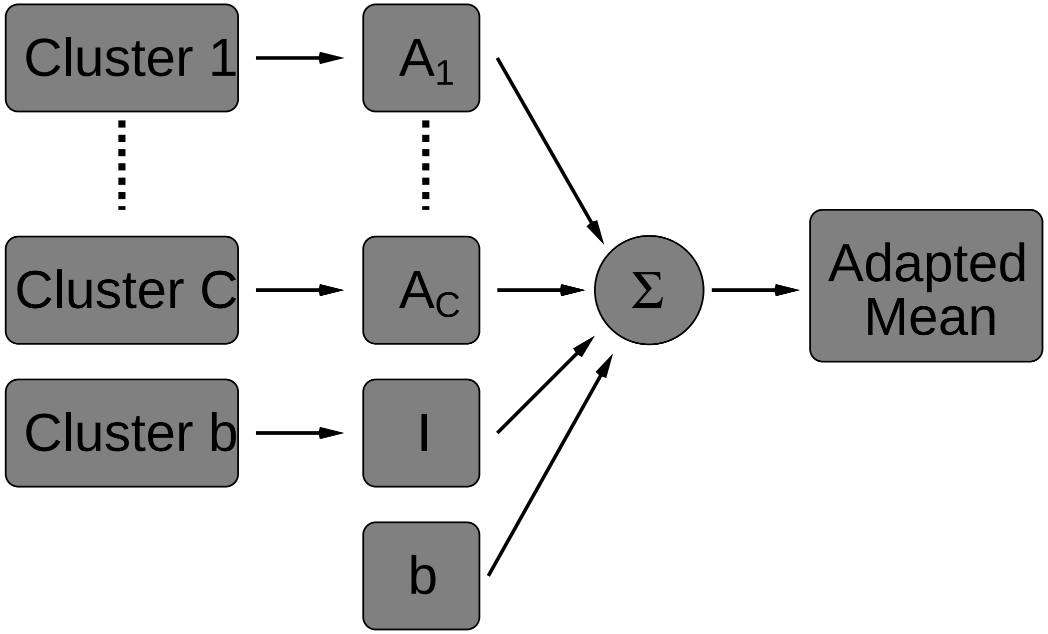
2. Goto (1) until converged.

Results

- Phone level classes $R = 46$
- Standard system $S = 1$
- Factored system $S = 39$ (one per dimension)

Classes	Assignment Scheme	Error Rate (%)	
		4 comp	12 comp
—	—	11.11	9.71
Global	Expert	10.34	8.87
Phone	Expert	10.04	8.86
	Standard	9.73	8.62
	Factored	9.48	8.42
State	Expert	9.20	8.84
Comp	—	9.22	9.98

Multiple-Cluster Transforms



Different systems described by:

- restricting number of clusters
- removing bias cluster
- restricting form of transform

Transform Descriptions

System	Transform Restrictions	Cluster Restrictions	Bias Cluster
SI	—	$C = 1$	No
SAT	—	$C = 1$	No
GD	$\mathbf{A}_i = \lambda_i \mathbf{I}$ $\lambda_i \in \{0, 1\}$, $\sum_i \lambda_i = 1$	$C = 2$	No
CAT	$\mathbf{A}_i = \lambda_i \mathbf{I}$	—	—
BSAT	—	$C = 1$	Yes
ESAT	—	—	—

General form for mean transform

$$\hat{\boldsymbol{\mu}}^{(sm)} = \mathbf{W}^{(s)} \boldsymbol{\xi}^{(m)}$$

where $\mathbf{W}^{(s)}$ is an $n \times p$ matrix and $\boldsymbol{\xi}^{(m)}$ is a $p \times 1$ vector, p is determined by the system.

Bias SAT

Interesting extension to SAT

$$\begin{aligned}\hat{\boldsymbol{\mu}}^{(sm)} &= \mathbf{A}^{(s)}\boldsymbol{\mu}^{(m)} + \boldsymbol{\mu}_{\mathbf{b}}^{(m)} + \mathbf{b}^{(s)} \\ &= \begin{bmatrix} \mathbf{A}^{(s)} & \mathbf{I} & \mathbf{b}^{(s)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}^{(m)} \\ \boldsymbol{\mu}_{\mathbf{b}}^{(m)} \\ 1 \end{bmatrix}\end{aligned}$$

Features

- Allows regression to be based around component specific point in acoustic space
- Bias cluster captures component aspects that are speaker independent (according to speaker transforms)
- No increase in number of transform parameters over standard SAT system

ML Training

Other than GD and CAT, modifications to standard SAT training.

1. Using *simple* models for each speaker to initialise clusters
2. Generate SI system and combine with initial *simple* clusters to initialise full clusters
3. Estimate transform for each training speaker given current estimate if multiple cluster model
4. Estimate mean of all clusters given variances and transforms
5. Estimate common (over clusters) variance given cluster means and transforms
6. Unless converged goto (3)

Initial Clusters

Two basic choices.

- **Speaker clustering**: using some distance measure cluster the simple speaker models. This yields a series of initial cluster assignments $\lambda_i \in \{0, 1\}$, $\sum_i \lambda_i = 1$.
- **Eigenvoices**: form a meta-vector from the means of each of the speaker models. Using either PCA or LDA reduce the dimensionality of the meta-vector to the required dimension. Project (or use ML) to form weights, λ_i associated with the reduced space.

Using both schemes simple ML estimates of the *complex* space.

For experiments here wher bias clusters are needed eigenvoice initialisation used. For no bias clusters speaker clustering is used. (Currently working on importance of inialisation for such systems).

Results

Only 2 cluster systems considered - ESAT uses no bias, CAT with bias.

System	Adapt	Error Rate (%)		
		H1 Dev	H1 Eval	Average
SI	—	9.38	10.01	9.71
	MLLR	8.24	9.24	8.77
SAT	MLLR	7.62	8.36	8.01
GD	GD	8.99	9.39	9.20
	ESAT	8.12	8.61	8.28
CAT	CAT	8.84	9.13	8.99
	ESAT	7.77	8.60	8.20
BSAT	BSAT	7.63	8.23	7.95
	ESAT	7.51	8.33	7.94
ESAT	ESAT	7.16	8.25	7.74

Other Interests

Current projects by students

- **Support vector machines**: are SVMs useful for speech - use of Fisher kernels.
- **Formant-like features**: direct anaysis of spectrum/peak picking as (hopefully) good/robust features.
- **Linear dynamical systems**: why can't we get these to work?