ICSLP 98

Speaker Clustering Using Direct Maximisation of the MLLR-Adapted Likelihood

Sue Johnson

Phil Woodland

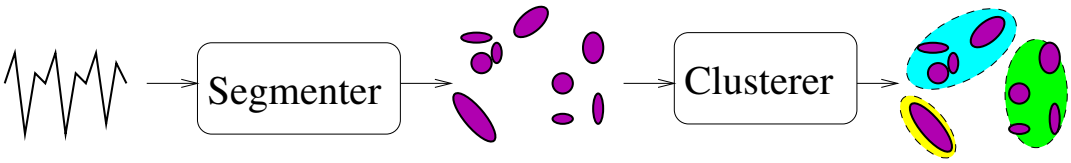
Email : sej28@eng.cam.ac.uk
pcw@eng.cam.ac.uk

http://svr-www.eng.cam.ac.uk/~sej28/papers/

Cambridge University Engineering Department

Motivation

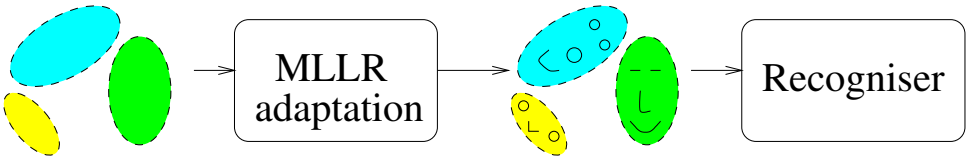
Recognition of Broadcast News Data



Pre-segmented audio: 1 speaker, 1 audio condition

MLLR allows adaptation to speaker/condition

Segments too small for robust adaptation



Possibly, want similar speakers/audio to go together

Ideally, want clusters specifically designed for MLLR

Cluster Architecture

parent node

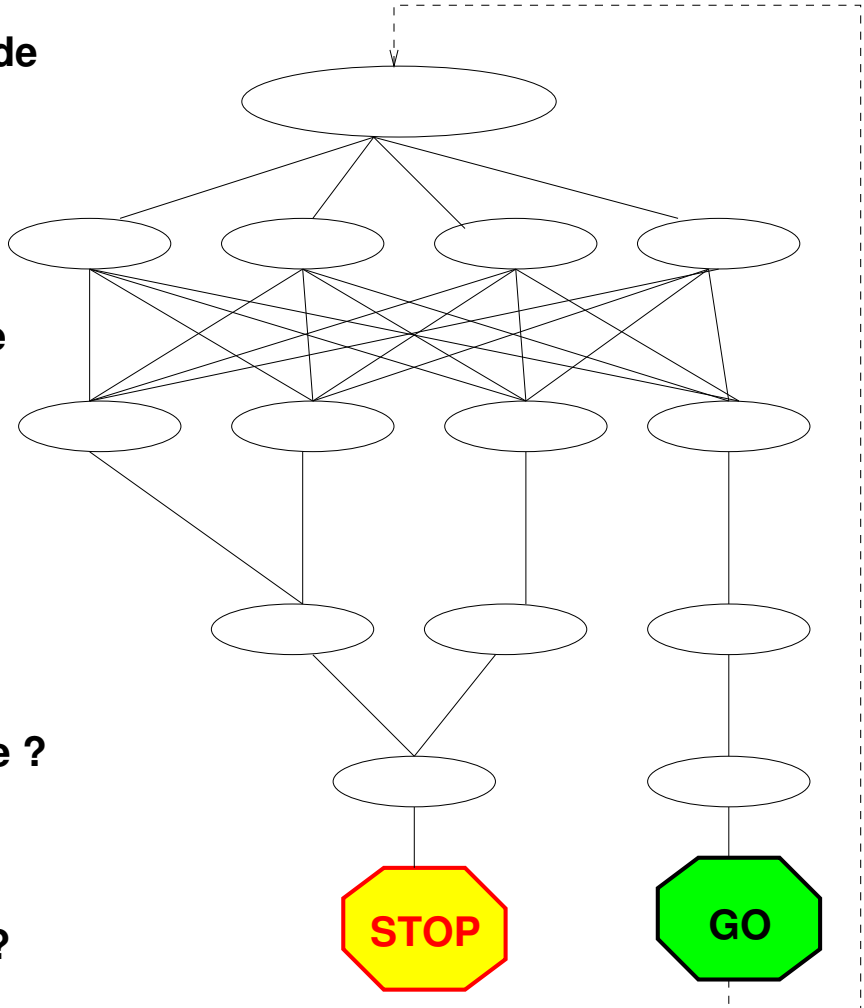
initialise

re-arrange

size ?

recombine ?

continue ?



Repeat until all nodes are leaf nodes

Covariance-Based Clustering

Represent segments with μ, Σ

Define distance between segments

$$AHM = \log\{tr(\Sigma_y \Sigma_x^{-1}) * tr(\Sigma_x \Sigma_y^{-1})\} - 2\log\{D\}$$

$$Div = 0.5 * tr\{\Sigma_x^{-1}\Sigma_y + \Sigma_y^{-1}\Sigma_x - 2I\} + 0.5 * (\mu_x - \mu_y)^T (\Sigma_x^{-1} + \Sigma_y^{-1})(\mu_x - \mu_y)$$

Initialise in chronological order

Split procedure for every parent node

```
Until no movers or max. iterations reached :
  Calculate mean and cov of each node
  Find distances between nodes and segments
  Assign segments to closest node
  Disperse segments if node too small
End
```

Recombine children if enclosed or overlap

Direct Maximisation (I)

Aim: Make clusters suitable for MLLR adaptation

Info: 2-level E-M algorithm

Guaranteed to increase likelihood of data

Split procedure for every parent node

```
Initialise Using Covariance Method
Until no movers or max. iterations reached:
  Calculate MLLR transform for each node
  Find likelihood of segments from transforms
  Assign segments to maximise likelihood
  Disperse segments if node too small
End
```

Direct Maximisation (II)

Problem Stopping criterion just occupancy

Tendency to always split

Need recombination of similar children

Define a cross-entropy like measure

$$\text{XProb}(i,k) = \log\left[\frac{p(i|i)\,p(k|k)}{p(i|k)\,p(k|i)}\right] = \log\left[\frac{\prod_{x=y} p(x|y)}{\prod_{x\neq y} p(x|y)}\right]$$

Recombine children with similar transform properties

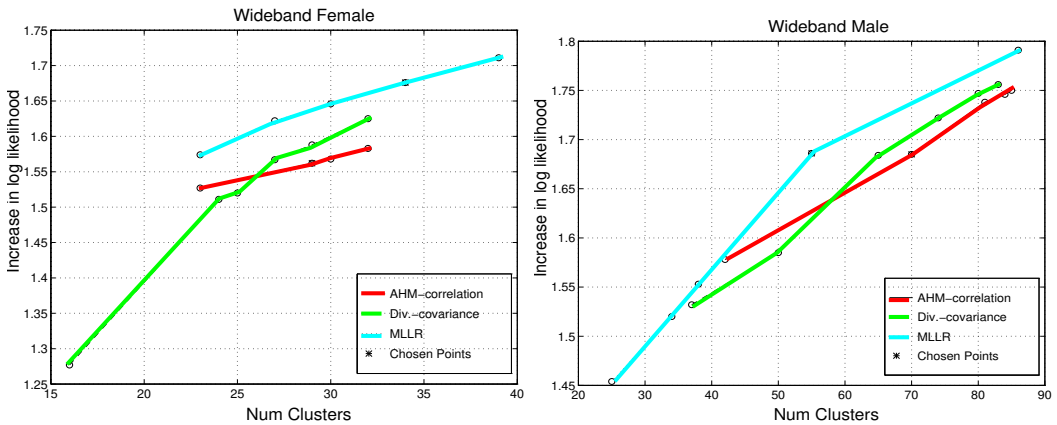
```
For each set of child nodes :
  Calculate XProb for each possible merge
  Find best merge (min XProb)
  Perform merge if XProb < threshold
  Flag merged as leaf node
End
```

Results (I)

Increase in log likelihood on dev97

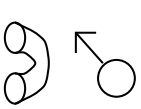
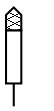
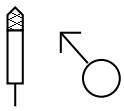
Clustering Method	NB Male	WB Male	WB Fem
CMU-based clusterer	1.774 (13)	1.596 (45)	1.598 (53)
Covariance based	1.808 (13)	1.746 (42)	1.596 (53)
+ recombination	1.850 (13)	1.761 (44)	1.617 (54)
Direct Maximisation	1.872 (14)	1.811 (47)	1.668 (57)
+ recombination	1.890 (14)	1.832 (46)	1.689 (56)

ROC varying recombination on dev97



Results (II)

Increase in log likelihood on eval97

Clustering Method				
Covariance based	1.344 (9)	1.586 (31)	2.238 (84)	2.142 (114)
+ recombination	1.269 (5)	1.543 (29)	2.209 (75)	2.109 (102)
Direct Maximisation	1.375 (10)	1.572 (29)	2.255 (83)	2.144 (117)
+ recombination	1.249 (4)	1.513 (21)	2.145 (52)	2.090 (96)

Reduction in WER on eval97

Clustering Method	Corr.	Sub.	Del.	Ins.	WER	#Cl
Unclustered	82.3	13.7	4.0	2.2	19.9	749
Covariance based	83.8	12.8	3.4	2.1	18.4	238
+ recombination	83.8	12.8	3.4	2.2	18.4	211
Direct Maximisation	83.9	12.7	3.4	2.1	18.3	239
+ recombination	83.9	12.7	3.4	2.1	18.2	173

Conclusions

- Covariance method OK and fast
- Likelihood of adapted data can be maximised directly
- Use a 2-step EM algorithm
- Add recombination using X-prob
- Increase in likelihood on dev97 and eval97
- Slight decrease in WER on eval97
- Reduced # clusters offers support for more transforms