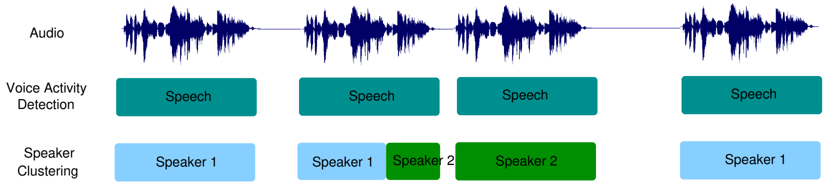
UNIVERSITY OF CAMBRIDGE

MLMI5: Audio Segmentation and Speaker Clustering

Mark Gales

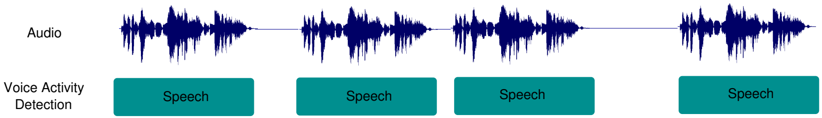
Lent 2019

Segmentation and Speaker Clustering



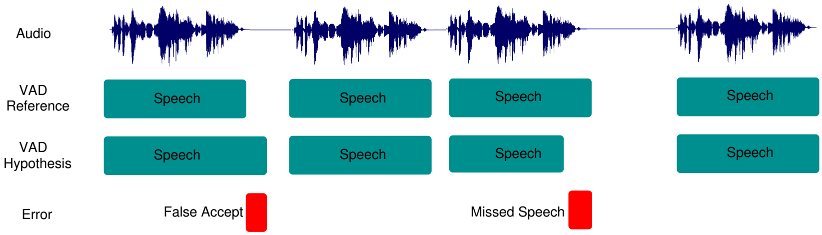
- Many tasks require voice activity detection
  - reduced computation load ("Hey Siri", "Alexa")
- Some tasks single audio stream, multiple speakers/conditions
  - broadcast media transcription
  - lecture transcription
  - YouTube captioning

Voice Activity Detection [13]



- Simple classification task: `speech/non-speech`
  - could run a full ASR system - yields words/silence
  - computationally expensive - possibly significant non-speech
- BUT not as trivial as it seems
  - wide-range of background (some structured) noise
  - possibly low signal-to-noise ratio (SNR)
  - channel/bandwidth conditions e.g. telephone/wide bandwidth

VAD Assessment

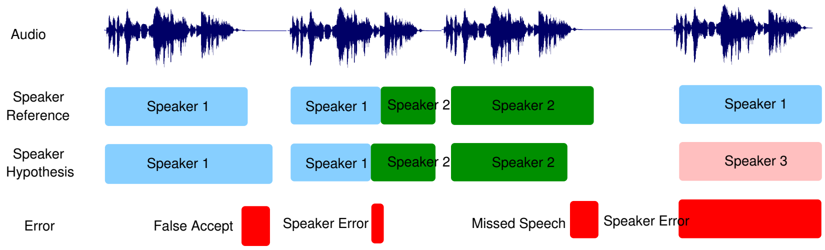


- VAD error is false accept plus missed speech
  - important to consider task - missed speech never recovered!
- Can also be assessed using ASR performance (or other task)

Example VAD Configuration: CUED MGB

- Training data (only lightly supervised data available)
  - 209 hours data of speech - selected with PMER=0%
  - 313 hours of intersegment silence - filtered using existing VAD
- DNN configuration (cross-entropy trained)
  - 40-dim filterbank features, ±27 frames of context
    contrast with ASR config ±5 frames of context
  - 6 hidden layers, 2 targets `speech`/`silence`
  - number of nodes $2200 \times 1000 \times 200^5 \times 2$
- Additional smoothing of classification (for final result)
  - change point detection and Iterative Agglomerative Clustering

Diarisation Assessment [11]

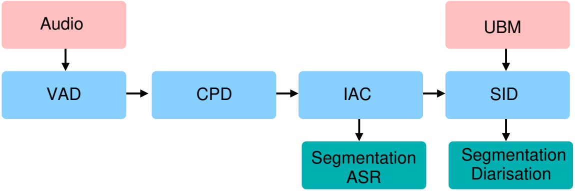


- Need to also group speech into "speakers"
- Three types of error:
  - missed speech (MS): same as VAD
  - false accept (FA): same as VAD
  - speaker error (SE): incorrect speaker label
- Scoring minimises error for speaker label mapping

ASR Adaptation and Diarisation

- Different clustering used for ASR and Diarisation
- ASR requires homogeneous clusters
  - adapts system to speaker/environment
  - each cluster requires minimum data for robust adaptation
- Diarisation penalises incorrect number of speakers
  - need to link same speaker in different acoustic conditions
  - single mapping from hypothesis to reference speakers IDs
- Often systems tuned to very different operating points

Speaker Clustering: CUED MGB Pipeline [4]



- Stages of CUED MGB Challenge system (fairly general)
  - Voice Activity Detection (VAD): speech/non-speech detection
  - Change Point Detection (CPD): speaker/environment changes
  - Iterative Agglomerative Clustering (IAC):homogeneous clusters
  - Speaker Identification (SID): refine clusters to only speakers
- Speaker segmentation task sometimes called diarisation

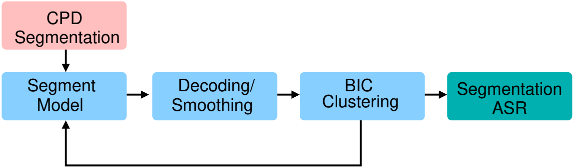
Change Point Detection (CPD)

- Range of options - CUED approach
  - parameterise audio with unnormalised features (MFCC)
  - train Gaussians (1 second either side of hypothesis point)
  - yields Gaussian distributions $p()$ and $q()$
  - measure symmetric KL divergence ($\texttt{KL2}()$):

$$\texttt{KL2}(p,q) = \frac{1}{2}\left(\mathcal{KL}(p||q) + \mathcal{KL}(q||p)\right)$$

  - select threshold above which hypothesise change point
- Select threshold to over-segment audio data
  - use IAC stage to merge clusters together

Iterative Agglomerative Clustering (IAC)



- Iterative clustering approach used:
  1. train model for each of current clusters
  2. decode speech audio data using cluster models
  3. smooth recognition output - new segments
  4. perform BIC clustering to form new clusters
- Single Gaussian segment models often used
  - diagonal or full covariance matrices

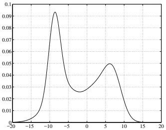
Bayesian Information Criterion (BIC) [10]

- Simple approximation to Bayesian approach

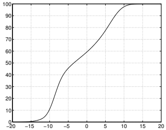
$$\begin{aligned}\log(p(\mathcal{D}|\mathcal{M})) &= \log\left(\int p(\mathcal{D}|\boldsymbol{\theta},\mathcal{M})p(\boldsymbol{\theta}|\mathcal{M})d\boldsymbol{\theta}\right)\\ &\approx \log(p(\mathcal{D}|\hat{\boldsymbol{\theta}},\mathcal{M})) - \frac{k}{2}\log(n) + R\end{aligned}$$

  - $\hat{\boldsymbol{\theta}}$: ML estimate of parameters $\boldsymbol{\theta}$
  - k: number of model parameters (size of $\boldsymbol{\theta}$)
  - n: number of samples in training data $\mathcal{D}$
  - R is the remainder (ignored)
- Often additional parameter α added
  - used to control model size (scales $k\log(n)$)
- Also possible to use minimum cluster size
  - useful when using clusters for speaker adaptation

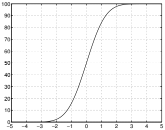
- Required to identify/cluster data from same speaker
- Need to remove environment/channel differences
  - CMN/CVN handle first and second moments
  - what about higher-order statistics?



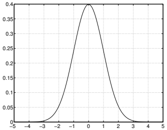
Source PDF



Source CDF



Target CDF



Target PDF

- Gaussianisation transforms data distribution to be Gaussian
  - normalises all moments of the distribution
  - for speaker clustering usually applied over 3 second window

Gaussianisation

- Approaches based on same concept (for dimension i)

$$\tilde{x}_i = \Phi^{-1}\left(\mathcal{C}(x_i)\right)$$

  - $\Phi()$ is the standard Gaussian CDF (inverse used)
  - $\mathcal{C}(x_i)$ is the CDF of the observed data distribution $p_{\text{obs}}()$ at x_i
  - use training data $\boldsymbol{x}_{1:T}$ to estimate data distribution
- Histogram equalisation: $h()$ is a step function

$$\mathcal{C}(x_i) = \int_{-\infty}^{x_i} p_{\text{obs}}(z)dz \approx \frac{1}{T}\sum_{t=1}^{T} h(x_i - x_{ti}) = \text{rank}(x_i)$$

- GMM-based Gaussianisation

$$\mathcal{C}(x_i) \approx \int_{-\infty}^{x_i} \sum_{m=1}^{M} c_m \mathcal{N}(z; \mu_i^{(m)}, \sigma_i^{(m)2})dz$$

Cross Likelihood Ratio (CLR)

- Distance criterion between AIC clusters required
  - range of approaches possible (including BIC)
  - CUED MGB system (and others) use Cross Likelihood Ratio
- CLR uses a Universal Background Model (UBM)
  - UBM is a large GMM used to represent all speakers

$$\mathtt{CLR}(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{n_i}\log\left(\frac{p(\mathcal{D}_i|\hat{\boldsymbol{\theta}}_j)}{p(\mathcal{D}_i|\boldsymbol{\theta}_{\mathtt{ubm}})}\right) + \frac{1}{n_j}\log\left(\frac{p(\mathcal{D}_j|\hat{\boldsymbol{\theta}}_i)}{p(\mathcal{D}_j|\boldsymbol{\theta}_{\mathtt{ubm}})}\right)$$

  - $\mathcal{D}_i$ data associated with cluster $\mathcal{C}_i$
  - $\hat{\boldsymbol{\theta}}_i$ ML estimate of model for data $\mathcal{D}_i$
  - n_i number of training samples in $\mathcal{D}_i$
  - $\boldsymbol{\theta}_{\mathtt{ubm}}$ UBM model parameters
- Merge clusters with highest CLR values

CUED MGB Diarisation Performance [4]

Series	DER (%)			
	MS	FA	SE	Tot
Sci-Fi drama	12.7	1.1	64.4	78.2
Sitcom	8.2	1.1	51.9	61.2
Documentary	1.9	0.2	10.8	12.9
TV-drama	6.4	1.0	16.3	23.7
Sports	5.7	1.6	39.9	47.1
Total	6.1	0.9	30.6	37.5

- CLR-based clustering from IAC: wide range of performance
    - challenging, diverse, shows have poor performance

Cross Episode Linking - Longitudinal Diarisation

- CLR can be used as the basis for linking
  - form an upper triangular matrix of CLR for all clusters

$$[\mathbf{D}]_{ij} = \mathtt{CLR}(\mathcal{C}_i, \mathcal{C}_j)$$

  - can get expensive for large numbers of episodes!
- Hierarchical merging of clusters then proceeds
  - CLR: update merged cluster parameters based on $\mathcal{D}_i, \mathcal{D}_j$
  - $\overline{\text{CLR}}$: distance to $\mathcal{C}_k$ becomes $([\mathbf{D}]_{ik} + [\mathbf{D}]_{jk})/2$
  - CLC: distance to $\mathcal{C}_k$ becomes $\min\{[\mathbf{D}]_{ik}, [\mathbf{D}]_{jk}\}$
- Threshold empirically set on development data

CUED MGB Longitudinal Diarisation Performance [4]

Linking Scheme	num Spkr		DER (%)	
	—	Link	—	Link
—	640	—	37.5	—
CLR	487	389	39.2	44.4
$\overline{\text{CLR}}$	533	426	38.9	43.9
CLC	599	473	37.9	42.7

- Linking over episodes degrades DER performance
  - two stage approaches probably not optimal
- Enables longitudinal speech recognition
  - interesting research direction ...

Speaker Representations

Variable Length Mapping

- Range of applications make use of speaker representations
  - speaker clustering
  - speaker recognition/verification
  - speaker adaptation
- All require a fixed-length representation
  - variable length sequence $\boldsymbol{x}_{1:T}^{(s)} \rightarrow$ speaker representation $\boldsymbol{\lambda}^{(s)}$

$$\boldsymbol{\lambda}^{(s)} = \phi(\boldsymbol{x}_{1:T}^{(s)})$$

  - in 4F10 already seen application using SVMs
  - makes use of Fisher Kernel
- Can make use of a UBM ($\boldsymbol{\theta}_{\tt ubm}$)
  - MAP adapt model to target speaker $\boldsymbol{\theta}_{\tt ubm}^{(s)}$ (see 4F10)

Fisher Kernel [8]

- From 4F10 lectures

$$\phi(\mathbf{x}_{1:T}) = \begin{bmatrix} \log(p(\mathbf{x}_{1:T}|\boldsymbol{\theta}^{(s)}_{\mathtt{ubm}})) - \log(p(\mathbf{x}_{1:T}|\boldsymbol{\theta}_{\mathtt{ubm}})) \\ \nabla_{\boldsymbol{\theta}} \log\left(p\left(\mathbf{x}_{1:T}|\boldsymbol{\theta}\right)\right)\Big|_{\boldsymbol{\theta}^{(s)}_{\mathtt{ubm}}} \end{bmatrix}$$

  - the first term is the standard GMM-based score
  - the second term is Fisher score for the speaker model
  - only derivatives wrt the mean parameters used
- If only the derivative part is used then

$$\phi(\mathbf{x}_{1:T}) = \begin{bmatrix} \sum_{t=1}^{T} P(1|\mathbf{x}_t, \boldsymbol{\theta}^{(s)}_{\mathtt{ubm}})\boldsymbol{\Sigma}^{(s1)-1}_{\mathtt{ubm}}(\mathbf{x}_t - \boldsymbol{\mu}^{(s1)}_{\mathtt{ubm}}) \\ \vdots \\ \sum_{t=1}^{T} P(\mathtt{M}|\mathbf{x}_t, \boldsymbol{\theta}^{(s)}_{\mathtt{ubm}})\boldsymbol{\Sigma}^{(sM)-1}_{\mathtt{ubm}}(\mathbf{x}_t - \boldsymbol{\mu}^{(sM)}_{\mathtt{ubm}}) \end{bmatrix}$$

- Fisher Information Matrix is sometimes used as a metric

Mean Super-Vector Kernel [1]

- Rather than taking derivative, it is possible to use parameters
  - consider the means of the speaker adapted UBM

$$\boldsymbol{\lambda}^{(s)} = \phi(\boldsymbol{x}^{(s)}_{1:T}) = \begin{bmatrix} \boldsymbol{\mu}^{(s1)}_{\texttt{ubm}} \\ \vdots \\ \boldsymbol{\mu}^{(sm)}_{\texttt{ubm}} \\ \vdots \\ \boldsymbol{\mu}^{(sM)}_{\texttt{ubm}} \end{bmatrix}$$

- Both this form and Fisher Kernel yield large spaces
  - if only means used $M \times d$ elements
  - originally used for SVM-based systems (see 4F10)
- Can we make the speaker information more compact?

Joint Factor Analysis (JFA) [6]

- The actual observed data is impacted by multiple factors
  - speaker (desired variability to model)
  - channel/session attributes (not desired)
- Decomposing the mean supervector yields

$$\boldsymbol{\lambda}^{(s)} = \boldsymbol{\mu}_{\mathtt{si}} + \mathbf{V}\boldsymbol{\lambda}^{(s)}_{\mathtt{sp}} + \mathbf{U}\boldsymbol{\lambda}^{(s)}_{\mathtt{ch}} + \mathbf{D}\boldsymbol{z}$$

  - $\mathbf{V}$ and $\mathbf{U}$ and loading matrices
  - $\boldsymbol{\mu}_{\mathtt{si}}$ is the speaker-independent mean
  - $\boldsymbol{\lambda}^{(s)}_{\mathtt{sp}}$ point in speaker-space (prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$)
  - $\boldsymbol{\lambda}^{(s)}_{\mathtt{ch}}$ point in channel/session-space (prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$)
  - $\mathbf{D}$ the noise matrix, $\boldsymbol{z}$ noise term (prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$)
- Effectively a large Gaussian distribution: typical dimensions
  - $\boldsymbol{\lambda}^{(s)}$: 20000; $\boldsymbol{\lambda}^{(s)}_{\mathtt{sp}}$: 300; $\boldsymbol{\lambda}^{(s)}_{\mathtt{ch}}$: 100
  - iterative training process - see paper

iVector Model Training [2]

- Identity Vector (iVector): simplify JFA merge speaker/channel

$$\boldsymbol{\lambda}^{(s)} = \boldsymbol{\mu}_{\mathtt{si}} + \mathbf{T}\boldsymbol{\lambda}_{\mathtt{sp}}^{(s)}$$

  - $\mathbf{T}$ is the total variability matrix
  - $\boldsymbol{\lambda}_{\mathtt{sp}}^{(s)}$ point in speaker-space (prior $\mathcal{N}(\mathbf{0},\mathbf{I})$)
- This is similar to Factor Analysis: use EM
  - unobserved: speaker $\boldsymbol{\lambda}_{\mathtt{sp}}$, component at t $P(m|\boldsymbol{\lambda}_{\mathtt{sp}}, \boldsymbol{x}_t^{(s)}\boldsymbol{\theta})$

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}) \quad = \quad & \sum_{s=1}^{S}\int p(\boldsymbol{\lambda}_{\mathtt{sp}}|\boldsymbol{\theta},\boldsymbol{x}_{1:T}^{(s)})\sum_{t=1}^{T}\sum_{m=1}^{M}P(m|\boldsymbol{\lambda}_{\mathtt{sp}},\boldsymbol{x}_t^{(s)},\boldsymbol{\theta}) \\ & \log\left(\mathcal{N}(\boldsymbol{x}_t^{(s)};\hat{\boldsymbol{\mu}}_{\mathtt{si}}^{(m)}+\hat{\mathbf{T}}^{(m)}\boldsymbol{\lambda}_{\mathtt{sp}},\hat{\boldsymbol{\Sigma}}^{(m)})\right)d\boldsymbol{\lambda}_{\mathtt{sp}}\end{aligned}$$

  - new model parameters $\hat{\boldsymbol{\theta}} = \left\{\ldots,\hat{\mathbf{T}}^{(m)},\hat{\boldsymbol{\mu}}_{\mathtt{si}}^{(m)},\hat{\boldsymbol{\Sigma}}^{(m)},\ldots\right\}$
  - for simplicity $P(m|\boldsymbol{\lambda}_{\mathtt{sp}},\boldsymbol{x}_t^{(s)},\boldsymbol{\theta})$ often fixed for training

iVector Extraction [5, 7, 3]

- At test-time iVector extracted using

$$\hat{\boldsymbol{\lambda}}^{(s)}_{\text{sp}} = \arg\max_{\boldsymbol{\lambda}_{\text{sp}}} \left\{ p(\boldsymbol{\lambda}_{\text{sp}} | \boldsymbol{x}^{(s)}_{1:T}, \boldsymbol{\theta}) \right\}$$

  - again EM is used to find iVector
- Model related to CAT and EigenVoices
  - point estimate of $\boldsymbol{\lambda}^{(s)}_{\text{sp}}$ used, rather than distribution
  - treated as part of the parameter estimation stage

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) \quad = \quad & \sum_{s=1}^{S}\sum_{m=1}^{M}\sum_{t=1}^{T} P(m|\boldsymbol{\lambda}^{(s)}_{\text{sp}}, \boldsymbol{x}^{(s)}_t \boldsymbol{\theta}) \Big[\log(P(\hat{\boldsymbol{\lambda}}^{(s)}_{\text{sp}})) \\ & + \log\left(\mathcal{N}(\boldsymbol{x}^{(s)}_t; \hat{\boldsymbol{\mu}}^{(m)}_{\text{si}} + \hat{\mathbf{T}}^{(m)}\hat{\boldsymbol{\lambda}}^{(s)}_{\text{sp}}, \hat{\boldsymbol{\Sigma}}^{(m)})\right)\Big]\end{aligned}$$

  - possible to factorise $\boldsymbol{\lambda}^{(s)}_{\text{sp}}$ (JFA) include orthogonality constraint

iVectors for Speaker Recognition

- Extract iVectors for all enrolled speakers, $\boldsymbol{\lambda}_{\text{sp}}^{(1)}, \ldots, \boldsymbol{\lambda}_{\text{sp}}^{(S)}$
  - extract for test speaker $\boldsymbol{\lambda}_{\text{sp}}$
  - need to select "closest" enrolled speaker
- For speed look at distances between iVectors

$$\hat{s} = \arg\min_{s} \left\{ d(\boldsymbol{\lambda}_{\text{sp}}, \boldsymbol{\lambda}_{\text{sp}}^{(s)}) \right\}$$

  - euclidean distance:

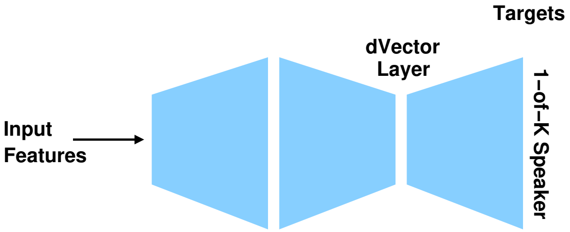
$$d(\boldsymbol{\lambda}_{\text{sp}}, \boldsymbol{\lambda}_{\text{sp}}^{(s)}) = ||\boldsymbol{\lambda}_{\text{sp}} - \boldsymbol{\lambda}_{\text{sp}}^{(s)}||^2$$

  - (-) cosine distance:

$$d(\boldsymbol{\lambda}_{\text{sp}}, \boldsymbol{\lambda}_{\text{sp}}^{(s)}) = -\frac{\boldsymbol{\lambda}_{\text{sp}}^{\mathsf{T}} \boldsymbol{\lambda}_{\text{sp}}^{(s)}}{\sqrt{\boldsymbol{\lambda}_{\text{sp}}^{\mathsf{T}} \boldsymbol{\lambda}_{\text{sp}} \boldsymbol{\lambda}_{\text{sp}}^{(s)\mathsf{T}} \boldsymbol{\lambda}_{\text{sp}}^{(s)}}}$$

    popular choice (empirically good!)

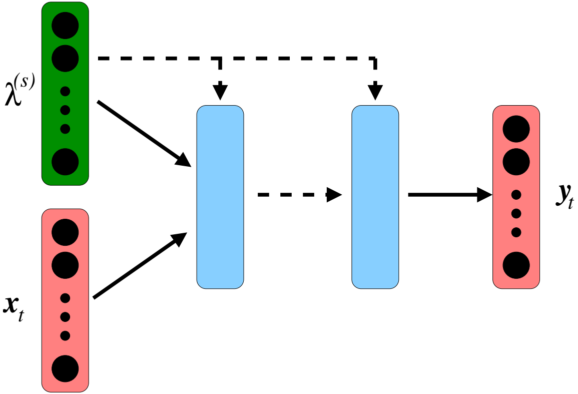
- Train vector to discriminate between speakers
  - related to bottleneck features for ASR



- Targets are a 1-of-K coding of speaker
  - wide window of features to yield good performance
- Simple approach used to handle temporal aspect of signal
  - $\boldsymbol{\lambda}_t^{(s)}$ is the vector for frames centered at time t

$$\boldsymbol{\lambda}^{(s)} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\lambda}_t^{(s)}$$

Speaker Representations for Adaptation



- Speaker representation can be used as auxiliary information
  - simple for of speaker adaptation
  - no initial hypothesis required
  - can be optionally be applied to other layers of network

[1] W. Campbell, D.Sturim, D. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. ICASSP*, 2006.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[3] M. J. F. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Transactions Speech and Audio Processing*, vol. 8, pp. 417–428, 2000.

[4] P. Karanasou, M. J. F. Gales, P. Lanchantin, X. Liu, Y. Qian, L. Wang, P. C. Woodland, and C. Zhang, "Speaker diarisation and longitudinal linking in multi-genre broadcast data," 2015.

[5] P. Karanasou, Y. Wang, M. Gales, and P. Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *Proceedings of Interspeech'14*, 2014.

[6] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. Audio, Speech & Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[7] R. Kuhn, P. Nguyen, J.-C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini, "Eigenvoices for speaker adaptation," in *Proceedings ICSLP*, 1998, pp. 1771–1774.

[8] C. Longworth and M. Gales, "Derivative and parametric kernels for speaker verification," in *Proceedings InterSpeech*, September 2007.

[9] J. W. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *2001: A Speaker Odyssey - The Speaker Recognition Workshop, Crete, Greece, June 18-22, 2001*, 2001, pp. 213–218.

[10] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1973.

[11] S. E. Tranter and D. A. Reynolds, "Speaker diarisation for broadcast news," in *Proc. Odyssey Speaker and Language Recognition Workshop*, Toledo, Spain, June 2004, pp. 337–344.

[12] E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4052–4056, 2014.

[13] L. Wang, C. Zhang, P. Woodland, M. Gales, P. Karanasou, P. Lanchantin, X. Liu, and Y. Qian, "Improved DNN-based segmentation for multi-genre broadcast audio," in *Proc. ICASSP'16*, 2016.