Outline

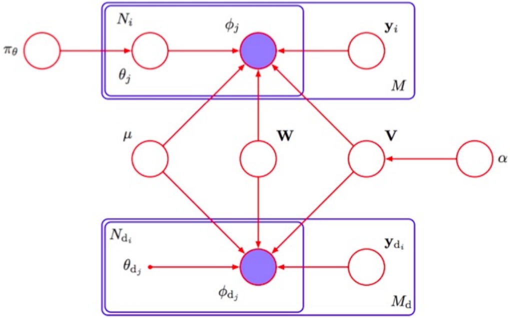
Outline

Domain adaptation

- Domain adaptation aims to adapt a system from a resource-rich domain to a resource-limited domain.
- Domain mismatch: systems perform very well in the domain (environment) for which they are trained; however, their performance suffers when the users use the systems in other domain.
- Domain Adaptation Challenge (DAC13) was introduced in 2013.
- The effect of domain mismatch on PLDA parameters is more pronounced
- **Popular approaches**:
  - Bayesian adaptation of PLDA models [Villalba and Lleida, 2014]
  - Unsupervised clustering of i-vectors for adapting covariance matrices of PLDA models [Shum et al., 2014, Garcia-Romero et al., 2014]
  - Inter-dataset variability compensation [Aronowitz, 2014, Kanagasundaram et al., 2015]

Bayesian Adaptation of PLDA (Villalba and Lleida, 2014)

- Use a generative model to generate out-of-domain labelled (θ_{d_j}) data and in-domain (target) unlabelled (θ_j) data.
- Unknown labels (θ_j) are modelled as latent variables



Generative model:

$$\phi_j = \mu + \mathbf{V}\mathbf{y}_i + \epsilon_j$$

Joint posterior of the latent variables:

$$P\left(\mathbf{Y}, \mathbf{Y}_{\mathrm{d}}, \theta, \pi_\theta, \mu, \mathbf{V}, \mathbf{W}, \alpha | \mathbf{\Phi}, \mathbf{\Phi}_{\mathrm{d}}\right) \approx$$

$$q\left(\mathbf{Y}, \mathbf{Y}_{\mathrm{d}}\right) q\left(\theta\right) q\left(\pi_\theta\right) \prod_{r=1}^{d} q\left(\tilde{\mathbf{v}}'_r\right) q\left(\mathbf{W}\right) q\left(\alpha\right)$$

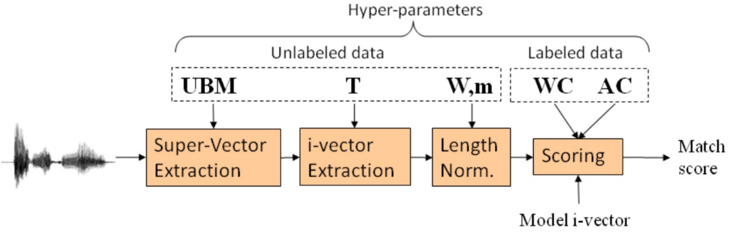
Unsupervised clustering of i-vectors (Shum et al., 2014)

Step 1 Use the within-class and between-class covariance matrices of out-of-domain data to compute a pairwise affinity matrix on unlabelled in-domain data.

Step 2 Use the affinity matrix to obtain hypothesized speaker clusters of in-domain data.

Step 3 Linearly interpolate the in-domain and out-of-domain covariance matrices to obtained the adapted matrices:

$$\boldsymbol{\Sigma}_{\text{adapt}} = \alpha_{\text{wc}} \boldsymbol{\Sigma}_{\text{in}} + (1 - \alpha_{\text{wc}} \boldsymbol{\Sigma}_{\text{out}})$$
$$\boldsymbol{\Phi}_{\text{adapt}} = \alpha_{\text{ac}} \boldsymbol{\Phi}_{\text{in}} + (1 - \alpha_{\text{ac}} \boldsymbol{\Phi}_{\text{out}})$$



Inter-dataset variability compensation (IDVC)

- IDVC aims at explicitly modeling dataset shift variability in the i-vector space and compensating it as a pre-processing cleanup step.
- No need to use in-domain data to adapt the PLDA model.

Step 1 Divide the development data into different subsets according to their sources, e.g., different LDC distributions of Switchboard.

Step 2 Estimate an inter-dataset variability subspace with the largest variability across the mean i-vectors of these subsets.

Step 3 Remove the variability of i-vectors in this subspace via nuisance attribute projection (NAP).

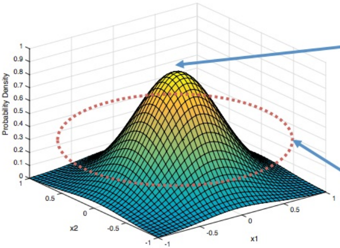
Outline

Short-utterance speaker verification

- Performance of i-vector/PLDA systems degrades rapidly when the systems are presented with short utterances or utterances with varying durations.
- These systems consider long and short utterances as equally reliable.
- However, the i-vectors of short utterances have much bigger posterior covariances.
- **Popular approaches**:
  - Uncertainty Propagation: Propagate the posterior covariances of i-vectors to PLDA [Kenny et al., 2013]
  - Full posterior distribution PLDA [Cumani et al., 2014]

Uncertainty propagation (Kenny et al., 2013)

- In i-vector extraction, besides the posterior mean of the latent variable (i-vector), we also have the posterior covariance matrix, which reflects the uncertainty of the i-vector estimate.

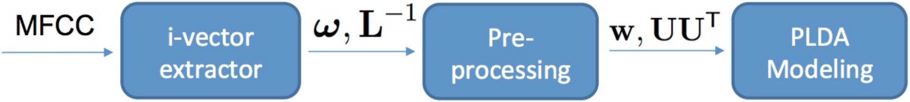


$$\boldsymbol{\omega} = \mathrm{cov}(\boldsymbol{\eta}, \boldsymbol{\eta}) \sum_{c=1}^{C} \mathbf{T}_c^{\mathsf{T}} \boldsymbol{\Sigma}_c^{-1} \tilde{\mathbf{f}}_c$$

$$\mathrm{cov}(\boldsymbol{\eta}, \boldsymbol{\eta}) = \mathbf{L}^{-1} = \left(\mathbf{I} + \sum_{c=1}^{C} N_c \mathbf{T}_c^{\mathsf{T}} \boldsymbol{\Sigma}_c^{-1} \mathbf{T}_c \right)^{-1}$$

$\mathbf{L}$ is the precision matrix of the posterior density
N_c is zero-order sufficient statistics with respect to UBM
$\tilde{\mathbf{f}}_c$ is first-order sufficient statistics with respect to UBM

Uncertainty propagation (Kenny et al., 2013)



$$\mathbf{w}_{ij} = \mathbf{m} + \mathbf{V}\mathbf{h}_i + \mathbf{U}_{ij}\mathbf{z}_{ij} + \epsilon_{ij}$$

- $\mathbf{U}_{ij}$ is the Cholesky decomposition of the posterior covariance matrix ($\mathbf{L}_{ij}^{-1}$) of the j-th i-vector by the i-th speaker.
- The intra-speaker covariance matrix becomes:

$$\mathrm{cov}(\mathbf{w}_{ij}, \mathbf{w}_{ij}|\mathbf{h}_i) = \mathbf{U}_{ij}\mathbf{U}_{ij}^{\mathsf{T}} + \boldsymbol{\Sigma}$$

  where $\mathbf{U}_{ij}\mathbf{U}_{ij}^{\mathsf{T}}$ changes from utterance to utterance, thus reflecting the reliability of the i-vector $\mathbf{w}_{ij}$.
- Scoring is computationally expensive. See [Lin and Mak, 2016] for a fast scoring algorithm.

Outline

Text-dependent SV using short utterances

- Very difficult for short-utterance text-dependent SV
- Text-dependent tasks with the uncertainty propagation version of i-vector/PLDA were unsatisfactory [Stafylakis et al., 2013].
- It is more natural to use HMMs rather than GMMs for text-dependent tasks. But HMMs require *local* hidden variables, which are difficult to handle because of data fragmentation.
- **Emergent approaches**:
  - Content matching [Scheffer and Lei, 2014]
  - Exploiting out-of-domain data via domain adaptation [Aronowitz and Rendel, 2014, Kenny et al., 2014]
  - y-vector and JFA backend [Kenny et al., 2015b]
  - I-vector backend [Kenny et al., 2015a]
  - Hidden suppervector backend [Kenny et al., 2016]
  - Use DNN/RNN to extract utterance-level features [Heigold et al., 2016, Bhattacharya et al., 2016, Zeinali et al., 2016]