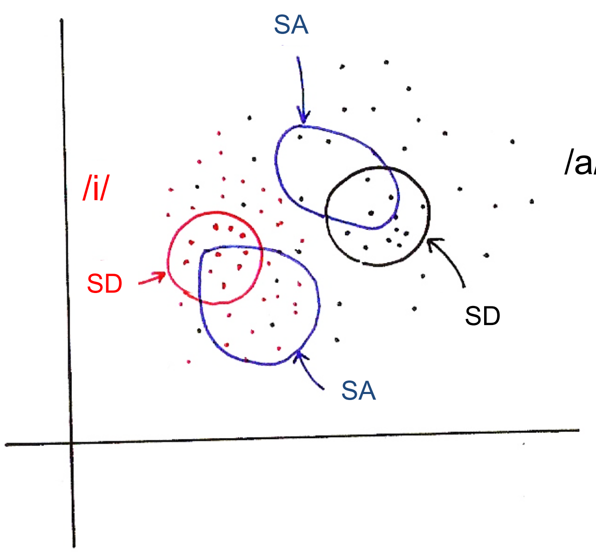
13.0 Speaker Variabilities: Adaption and Recognition

References:
1. 9.6 of Huang
2. " Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains", IEEE Trans. on
3. " Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models", Computer Speech and Language, Vol.9 ,1995
4. Jolliffe, " Principal Component Analysis ", Springer-Verlag, 1986
5. " Rapid Speaker Adaptation in Eigenvoice Space", IEEE Trans. on Speech and Audio Processing, Nov 2000
6. " Cluster Adaptive Training of Hidden Markov Models", IEEE Trans. on Speech and Audio Processing, July 2000
7. " A Compact Model for Speaker-adaptive Training", International Conference on Spoken Language Processing, 1996
8. " A Tutorial on Text-independent Speaker Verification", EURASIP Journal on Applied Signal Processing 2004
9. "An Overview of Text-independent Speaker Recognition: from Features to Supervectors", Speech Communication, Jan 2010
10. "VoxCeleb2: Deep Speaker Recognition", Interspeech 2018

Speaker Dependent/Independent/Adaptation

- **Speaker Dependent (SD)**
  - trained with and used for 1 speaker only, requiring huge quantity of training data, best accuracy
  - practically infeasible
- **Multi-speaker**
  - trained for a (small) group of speakers
- **Speaker Independent (SI)**
  - trained from large number of speakers, each speaker with limited quantity of data
  - good for all speakers, but with relatively lower accuracy
- **Speaker Adaptation (SA)**
  - started with speaker independent models, adapted to a specific user with limited quantity of data (adaptation data)
  - technically achievable and practically feasible
- **Supervised/Unsupervised Adaptation**
  - supervised: text (transcription) of the adaptation data is known
  - unsupervised: text (transcription) of the adaptation data is unknown, based on recognition results with speaker-independent models, may be performed iteratively
- **Batch/Incremental/On-line Adaptation**
  - batch: based on a whole set of adaptation data
  - incremental/on-line: adapted step-by-step with iterative re-estimation of models e.g. first adapted based on first 3 utterances, then adapted based on next 3 utterances or first 6 utterances,...

Speaker Dependent/Independent/Adaptation



MAP (Maximum A Posteriori) Adaptation

- **Given Speaker-independent Model set $\Lambda=\{\lambda_i=(A_i, B_i, \pi_i), i=1, 2,...M\}$ and A set of Adaptation Data $\overline{O} = (o_1, o_2,...o_t,...o_T)$ for A Specific Speaker**

$$\Lambda^* = \underset{\Lambda}{\arg\max}\ \text{Prob}[\Lambda|\overline{O}] = \underset{\Lambda}{\arg\max}\ \frac{\text{Prob}[\overline{O}|\Lambda]\text{Prob}[\Lambda]}{\text{Prob}[\overline{O}]} = \underset{\Lambda}{\arg\max}\ \text{Prob}[\overline{O}|\Lambda]\text{Prob}[\Lambda]$$

- **With Some Assumptions on the Prior Knowledge Prob $[\Lambda]$ and some Derivation (EM Theory)**
  - example adaptation formula

$$\mu^*_{jk} = \frac{\tau_{jk}\mu_{jk} + \Sigma_{t=1}^{T}[\gamma_t(j,k)o_t]}{\tau_{jk} + \Sigma_{t=1}^{T}\gamma_t(j,k)}$$

μ_{jk} : mean of the k-th Gaussian in the j-th state for a certain λ_i

μ^*_{jk} : adapted value of μ_{jk}

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\Sigma_{j=1}^{N}\alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk}N(o_t;\mu_{jk},U_{jk})}{\Sigma_{m=1}^{L}c_{jm}N(o_t;\mu_{jm},U_{jm})}\right]$$

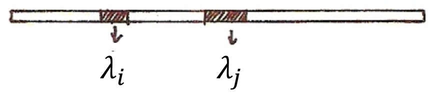
$$\uparrow$$

$$\gamma_t(j) = P(q_t = j|\overline{O},\lambda_i)$$

τ_{jk}: a parameter having to do with the prior knowledge about μ_{jk} may have to do with number of samples used to train μ_{jk}

  - a weighted sum shifting μ_{jk} towards those directions of o_t (in j-th state and k-th Gaussian) larger τ_{jk} implies less shift
- **Only Those Models with Adaptation Data will be Modified, Unseen Models remain Unchanged — MAP Principle**
  - good with larger quantity of adaptation data
  - poor performance with limited quantity of adaptation data

MAP Adaptation

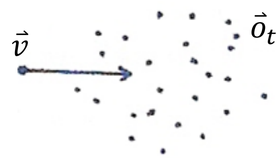


$$\frac{a\vec{v}+b\vec{o}}{a+b} = \lambda\vec{v}+(1-\lambda)\vec{o}$$

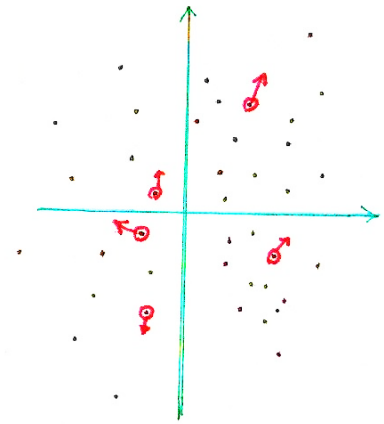
$$\Rightarrow \frac{a\vec{v}+(\sum_i b_i)\vec{o}}{a+(\sum_i b_i)}$$

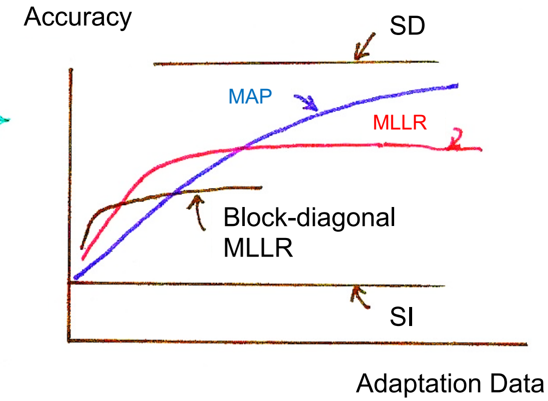
$$\Downarrow$$

$$\frac{a\vec{v}+(\sum_t b_t\vec{o}_t)}{a+(\sum_t b_t)}$$



MAP Adaptation





Maximum Likelihood Linear Regression (MLLR)

- **Divide the Gaussians (or Models) into Classes $C_1, C_2, \ldots C_L$, and Define Transformation-based Adaptation for each Class**

  $\mu^*_{jk} = A\mu_{jk}+b$, μ_{jk} : mean of the k-th Gaussian in the j-th state

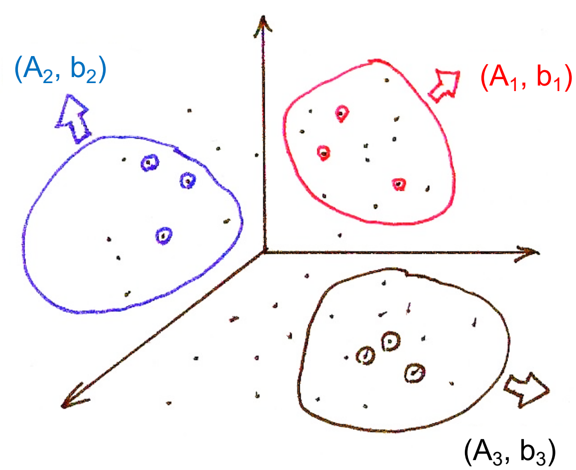
  – linear regression with parameters A, b estimated by maximum likelihood criterion

  $[A_i, b_i] = \arg\max_{A,b} \text{Prob}[\bar{O}|\Lambda, A_i, b_i]$ for a class C_i

  A_i, b_i estimated by EM algorithm

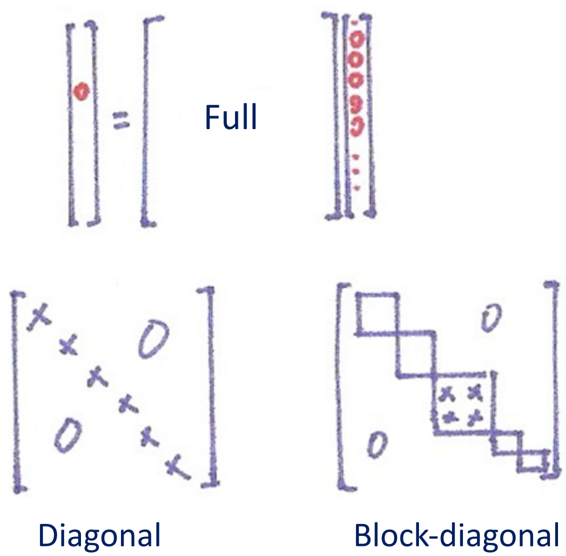
  – All Gaussians in the same class updated with the same A_i, b_i: parameter sharing, adaptation data sharing
  – unseen Gaussians (or models) can be adapted as well
  – A_i can be full matrices, or reduced to diagonal or block-diagonal to have less parameters to be estimated
  – faster adaptation with much less adaptation data needed, but saturated at lower accuracy with more adaptation data due to the less precise modeling
- **Clustering the Gaussians (or Models) into L Classes**
  – too many classes requires more adaptation data, too less classes becomes less accurate
  – basic principle: Gaussian (or models) with similar properties and " just enough" data form a class
  – data-driven (e.g. by Gaussian distances) primarily, knowledge driven helpful
- **Tree-structured Classes**
  – the node including minimum number of Gaussians (or models) but with adequate adaptation data is a class
  – dynamically adjusting the classes as more adaptation data are observed
- **Feature-based MLLR (fMLLR)**

MLLR

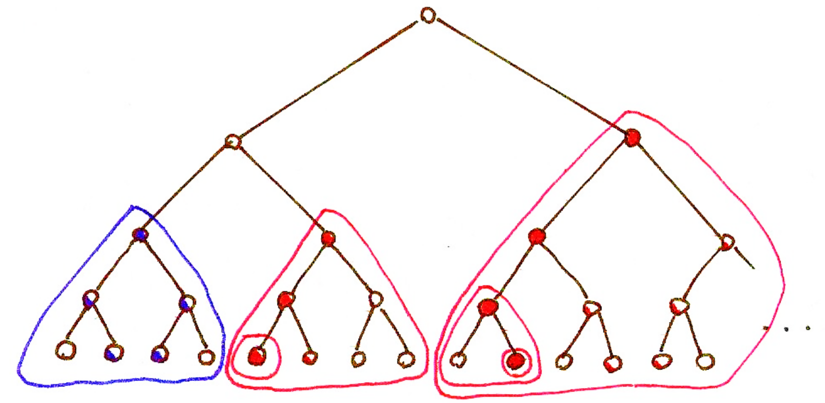


MLLR

$\vec{y} = A\vec{x}$



MLLR



Principal Component Analysis (PCA)

- **Problem Definition:**
  - for a zero mean random vector $\boldsymbol{x}$ with dimensionality $\boldsymbol{N}$, $x \in R^N$, $E(x)=0$, iteratively find a set of k ($k \le N$) orthonormal basis vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_k\}$ so that
    (1) *var* $(\boldsymbol{e}_1^T \boldsymbol{x})$=*max* *(x has maximum variance when projected on* e_1*)*
    (2) *var* $(\boldsymbol{e}_i^T \boldsymbol{x})$=*max*, subject to $\boldsymbol{e}_i \perp \boldsymbol{e}_{i-1} \perp \ldots\ldots \perp \boldsymbol{e}_1$, $2 \le i \le k$
    (x has next maximum variance when projected on e_2*, etc.)*
- **Solution: $\{e_1, e_2, \ldots, e_k\}$ are the eigenvectors of the covariance matrix Σ for x corresponding to the largest k eigenvalues**
  - new random vector $\boldsymbol{y} \in \boldsymbol{R}^k$ **:** the projection of x onto the subspace spanned by $\boldsymbol{A}=[\boldsymbol{e}_1\ \boldsymbol{e}_2 \ldots\ldots \boldsymbol{e}_k]$, $\boldsymbol{y}=\boldsymbol{A}^T\boldsymbol{x}$
  - a subspace with dimensionality $k \le N$ such that when projected onto this subspace, y is "closest" to x in terms of its "randomness" for a given k
  - var $(e_i^T x)$ is the eigenvalue associated with e_i
- **Proof**
  - var $(e_1^T x) = e_1^T E(x x^T) e_1 = e_1^T \Sigma e_1 = \max$, subject to $|e_1|^2=1$
  - using Lagrange multiplier

    $J(e_1) = e_1^T E(x x^T) e_1 - \lambda(|e_1|^2 - 1)$, $\frac{\partial J(e_1)}{\partial e_1} = 0$

    $\Rightarrow E(xx^T)\, e_1 = \lambda_1 e_1$, $var(e_1^T x) = \lambda_1 = \max$

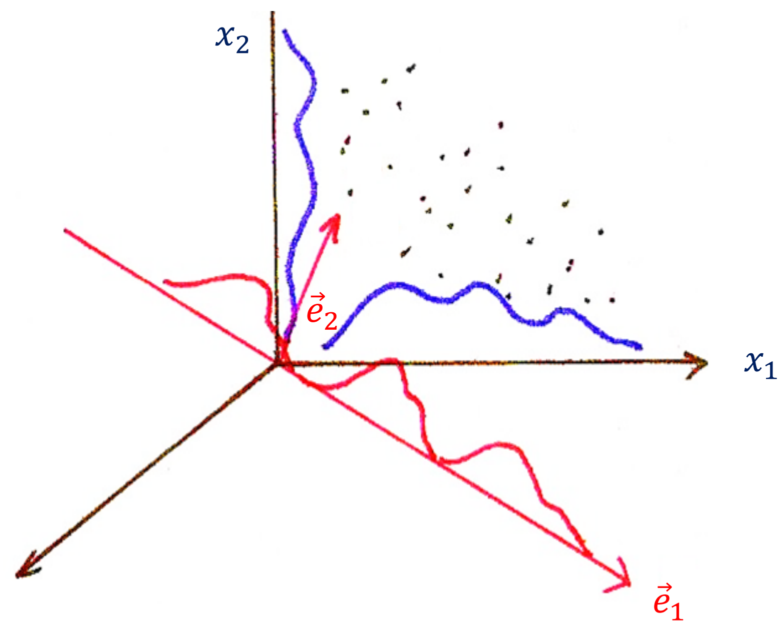
  - similar for e_2 with an extra constraint $e_2^T e_1 = 0$, etc.

$Av = u$

$Av = \lambda v$

eigenvector eigenvalue

PCA



PCA

$$[\cdots\ e_{1k}^T\ \cdots]\begin{bmatrix}\vdots\\ x_k\\ \vdots\end{bmatrix} = \vec{e}_1\cdot\vec{x}$$

$$= \underbrace{|\vec{e}_1|}_{\substack{\parallel\\ 1}}|\vec{x}|\ \cos\theta$$

$$\vec{y} = A^T\vec{x} = \begin{bmatrix}\vec{e}_1^{\,T}\\ \vec{e}_2^{\,T}\\ \vdots\\ \vec{e}_k^{\,T}\end{bmatrix}\vec{x}$$

Basic Problem 3 (P.35 of 4.0)

$$\bar{U} = \begin{array}{cc} & m \\ l & \begin{bmatrix} & \vdots & \\ \cdots & \bar{u}_{lm} & \cdots \\ & \vdots & \end{bmatrix}\end{array} = E(\begin{bmatrix}x_1-\bar{x}_1\\ x_2-\bar{x}_2\\ \vdots\\ \vdots\end{bmatrix}[x_1-\bar{x}_1, x_2-\bar{x}_2,\cdots])$$

$$\bar{u}_{lm} = E[(x_l-\bar{x}_l)(x_m-\bar{x}_m)]$$

PCA

$$\begin{bmatrix}\vec{e}_1\vec{e}_2\cdots\vec{e}_k & \cdots & \vec{e}_N\end{bmatrix}\begin{bmatrix}\lambda_1 & & & & & 0\\ & \lambda_2 & & & & \\ & & \ddots & & & \\ & & & \lambda_k & & \\ & & & & \ddots & \\ 0 & & & & & \lambda_N\end{bmatrix}\begin{bmatrix}\vec{e}_1^{\,T}\\ \vec{e}_2^{\,T}\\ \vdots\\ \vec{e}_k^{\,T}\\ \vdots\\ \vec{e}_N^{\,T}\end{bmatrix}$$

$\approx \Sigma$

$= \Sigma$

$= E[x\,x^T]$

$= E[(x-\bar{x})(x-\bar{x})^T]$

$$\sum \underbrace{\vec{e}_i}_{\text{eigenvector}} = \underbrace{\lambda_i}_{\text{eigenvalue}}\ \vec{e}_i$$

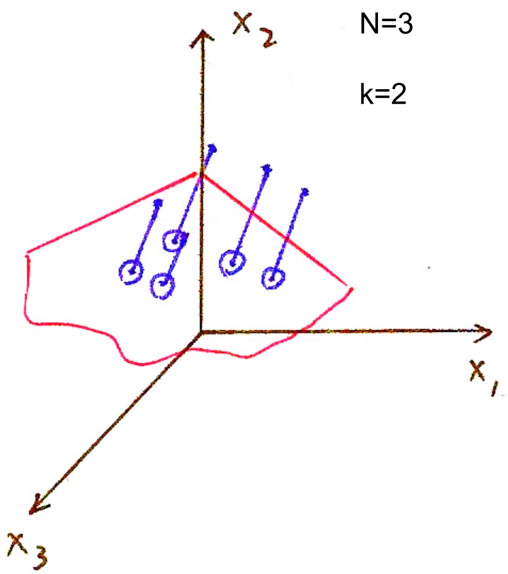
$$Av = u$$
$$Av = \lambda v$$

eigenvector eigenvalue

PCA

$$\vec{y} = A^T\vec{x}$$

PCA



N=3

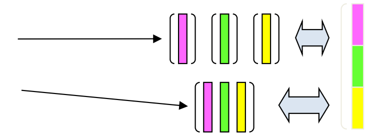
k=2

Eigenvoice

- **A Supervector x constructed by concatenating all relevant parameters for the speaker specific model of a training speaker**
  - concatenating the mean vectors of Gaussians in the speaker-dependent phone models
  - concatenating the columns of A, b in MLLR approach
  - x has dimensionality N (N = 5,000 × 3 × 8 × 40 = 4,800,000 for example)

    · SD model mean parameters (m)

    · transformation parameters (A, b)



- **A total of L (L = 1,000 for example) training speakers gives L supervectors $x_1, x_2, \ldots x_L$**
  - $x_1, x_2, x_3 \ldots\ldots x_L$ are samples of the random vector x
  - each training speaker is a point (or vector) in the space of dimensionality N
- **Principal Component Analysis (PCA)**
  - $x' = x - E(x)$, $\Sigma = E(x' x'^T)$,

    $\Sigma \approx [e_1, e_2 \ldots e_K][\lambda_i][e_1, e_2 \ldots e_k]^T$,$[\lambda_i]$: diagonal with λ_i as elements

    $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots\ldots \boldsymbol{e}_k\}$: eigenvectors with maximum eigenvalues $\lambda_1 > \lambda_2 \ldots > \lambda_k$

    k is chosen such that λ_j, j>k is small enough (k=250 or 50 for example)

Eigenvoice

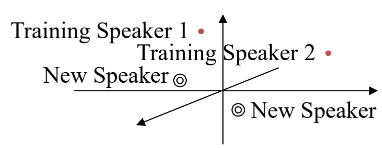
- **Principal Component Analysis (PCA)**
  - $x' = x - E(x)$, $\Sigma = E(x' x'^T)$,

    $\Sigma \approx [e_1, e_2 \ldots e_K][\lambda_i][e_1, e_2 \ldots e_k]^T$,$[\lambda_i]$: diagonal with λ_i as elements

    $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots\ldots \boldsymbol{e}_k\}$: eigenvectors with maximum eigenvalues $\lambda_1 > \lambda_2 \ldots > \lambda_k$

    k is chosen such that λ_j, j>k is small enough (k=50 for example)
- **Eigenvoice Space: spanned by $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots\ldots \boldsymbol{e}_k\}$**
  - each point (or vector) in this space represents a whole set of tri-phone model parameters
  - $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots\ldots \boldsymbol{e}_k\}$ represents the most important characteristics of speakers extracted from huge quantity of training data by large number of training speakers
  - each new speaker as a point (or vector) in this space, $y = \sum_{i=1}^{k} a_i e_i$
  - a_i estimated by maximum likelihood principle (EM algorithm)

    $\bar{a}^* = \arg\max_{\bar{a}} \mathrm{Prob}[\bar{O} | \sum_{i=1}^{k} a_i e_i]$

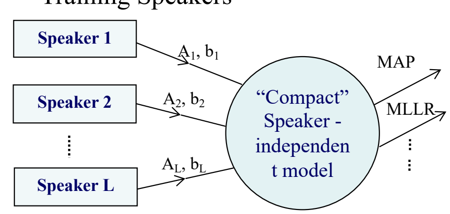


- **Features and Limitations**
  - only a small number of parameters $a_1 \ldots a_k$ is needed to specify the characteristics of a new speaker
  - rapid adaptation requiring only very limited quantity of training data
  - performance saturated at lower accuracy (because too few free parameters)
  - high computation/memory/training data requirements

Speaker Adaptive Training (SAT) and Cluster Adaptive Training (CAT)

- **Speaker Adaptive Training (SAT)**
  - trying to decompose the phonetic variation and speaker variation
  - removing the speaker variation among training speakers as much as possible
  - obtaining a “compact” speaker-independent model for further adaptation
  - y=Ax+b in MLLR can be used in removing the speaker variation
- **Clustering Adaptive Training (CAT)**
  - dividing training speakers into R clusters by speaker clustering techniques
  - obtaining mean models for all clusters(may include a mean-bias for the “compact” model in SAT)
  - models for a new speaker is interpolated from the mean vectors
- **Speaker Adaptive Training (SAT)**

  Training Speakers

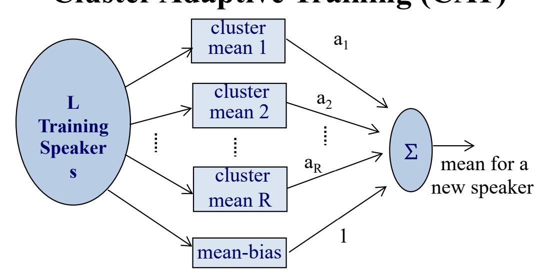


Original SI: $\Lambda^* = \arg\max_{\Lambda} \mathrm{Prob}(\bar{o}_{1,2 \ldots L} | \Lambda)$

SAT: $[\Lambda_c^*, (A,b)^*_{1,\ldots L}] = \arg\max_{\Lambda_c, (A,b)_{1,\ldots L}} \mathrm{Prob}(\bar{o}_{1,2 \ldots L} | \Lambda_c, (A,b)_{1,\ldots L})$

EM algorithm used

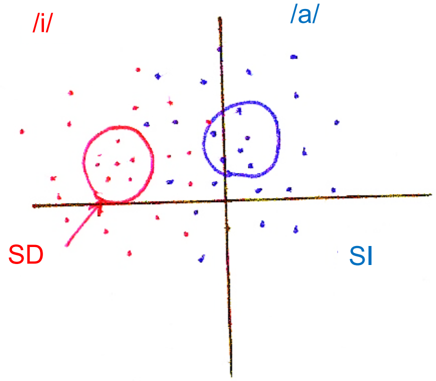
- **Cluster Adaptive Training (CAT)**



$m^* = \sum_{i=1}^{R} a_i m_i + m_b$, m_i: cluster mean i, m_b: mean-bias

a_i estimated with maximum likelihood criterion

SAT



$$\vec{y} = A\vec{x} + \vec{b}$$

$$\vec{x} = A^{-1}\,\vec{y} - A^{-1}\vec{b}$$

Speaker Recognition/Verification

- **To recognize the speakers rather than the content of the speech**
  - phonetic variation/speaker variation
  - speaker identification: to identify the speaker from a group of speakers
  - speaker verification: to verify if the speaker is as claimed
- **Gaussian Mixture Model (GMM)**

  $\lambda_i=\{(w_j, \mu_j, \Sigma_j), j=1,2,...M\}$ for speaker i

  for $\overline{O} = o_1 o_2 ... o_t ... o_T$, $b_i(o_t) = \sum_{j=1}^{M} w_j N(o_t; \mu_j, \Sigma_j)$
  - maximum likelihood principle

    $i^* = \arg\max_i \text{Prob}(\overline{O}|\lambda_i)$
- **Feature Parameters**
  - those carrying speaker characteristics preferred
  - MFCC
  - MLLR coefficients A_i, b_i, eigenvoice coefficients a_i, CAT coefficients a_i
- **Speaker Verification**
  - text dependent: higher accuracy but easily broken
  - text independent
  - likelihood ratio test

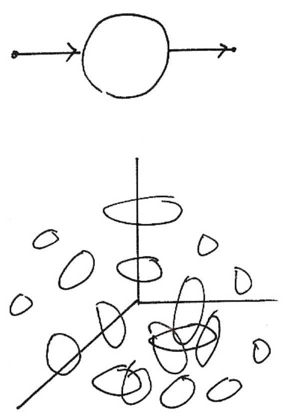
    $$\rho(\overline{O}; \lambda_i) = \frac{p(\overline{O}|\lambda_i)}{p(\overline{O}|\overline{\lambda}_i)} > th$$

    $\overline{\lambda}_i$: background model or anti - model for speaker i, trained by other speakers, competing speakers, or speaker - independent model

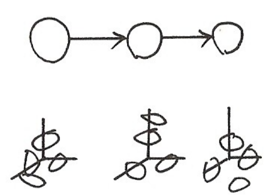
    th : threshold adjusted by balancing missing/false alarm rates and ROC curre
  - speech recognition based verification

Speaker Recognition

Gaussian Mixture Model (GMM)



HMM



Likelihood Ratio Test

- **Detection Theory— Hypothesis Testing/Likelihood Ratio Test**
  - 2 Hypotheses: H_0, H_1 with prior probabilities: $P(H_0)$,$P(H_1)$

    observation: X with probabilistic law: $P(X|H_0)$, $P(X|H_1)$
  - MAP principle

    choose H_0 if $P(H_0|X) > P(H_1|X)$

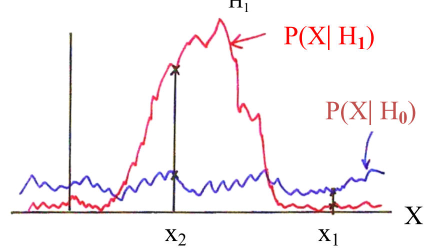
    choose H_1 if $P(H_1|X) > P(H_0|X)$

    $$\Rightarrow \frac{P(H_0|X)}{P(H_1|X)} \underset{H_1}{\overset{H_0}{\gtrless}} 1$$
  - Likelihood Ratio Test

    $P(H_i|X) = P(X|H_i)P(H_i)/P(X)$, i=0,1

    $$\Rightarrow \frac{P(X|H_0)}{P(X|H_1)} \underset{H_1}{\overset{H_0}{\gtrless}} \frac{P(H_1)}{P(H_0)} = Th$$

    likelihood ratio-Likelihood Ratio Test



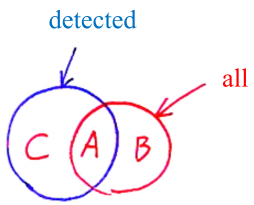
  - Type I error: missing (false rejection)

    Type II error: false alarm (false detection)

    false alarm rate, false rejection rate, detection rate, recall rate, precision rate

    Th: a threshold value adjusted by balancing among different performance rates

Receiver Operating Characteristics (ROC) Curve



Missing rate $= \frac{B}{A+B} = 1 - \frac{A}{A+B}$ recall

False Alarm rate $= \frac{C}{A+C} = 1 - \frac{A}{A+C}$ precision