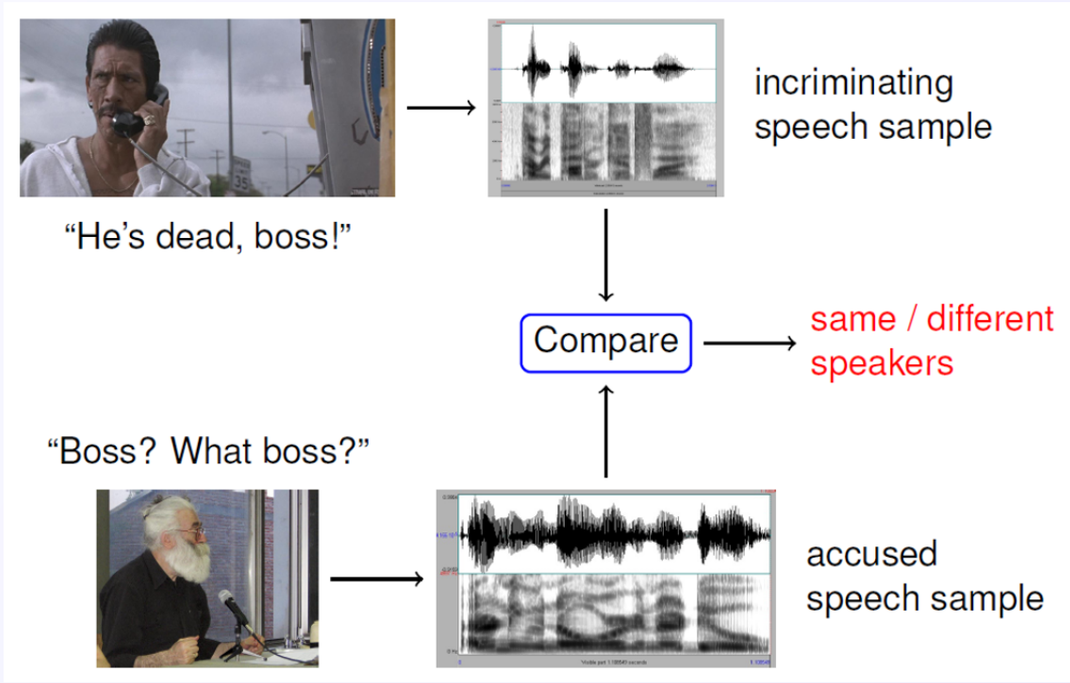
COM4511/COM6511 - Speech Technology

Lecture 17
Speaker Recognition



Thomas Hain
t.hain@sheffield.ac.uk
Spring Semester

SPANDH

The University Of Sheffield.

Overview

- Introduction
- PCA and subspace modelling
- Supervectors
- IVectors
- NNs (other verctors)
- SI systems

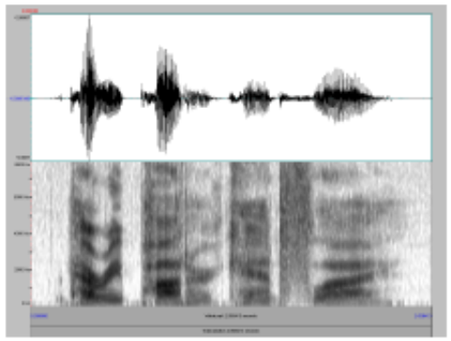
.. one of the core speech technology applications

- Various applications ...
  - Speaker recognition : Recognising the identity of speakers
  - Diarisation : Recognising who speaks when
  - Voice spoofing detection : Detecting authenticity of the speakers – impersonation, replay, speech synthesis, voice conversion
  - Language recognition : Identify the language spoken

- General idea: Learn arbitrary groupings on acoustic and linguistic features of speech

Speaker verification

"He's dead, boss!"

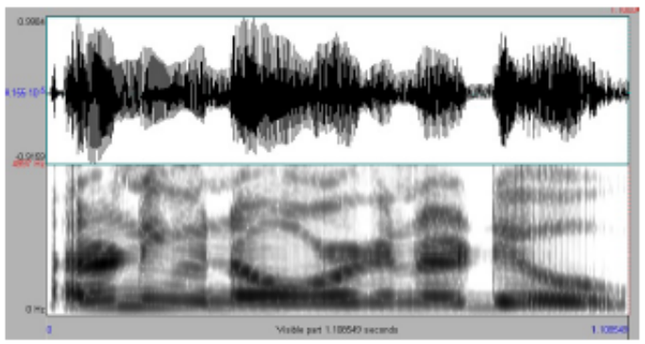


incriminating speech sample

Compare

same / different speakers

"Boss? What boss?"



accused speech sample

[Bruenner 2014]

Evaluating speaker verification

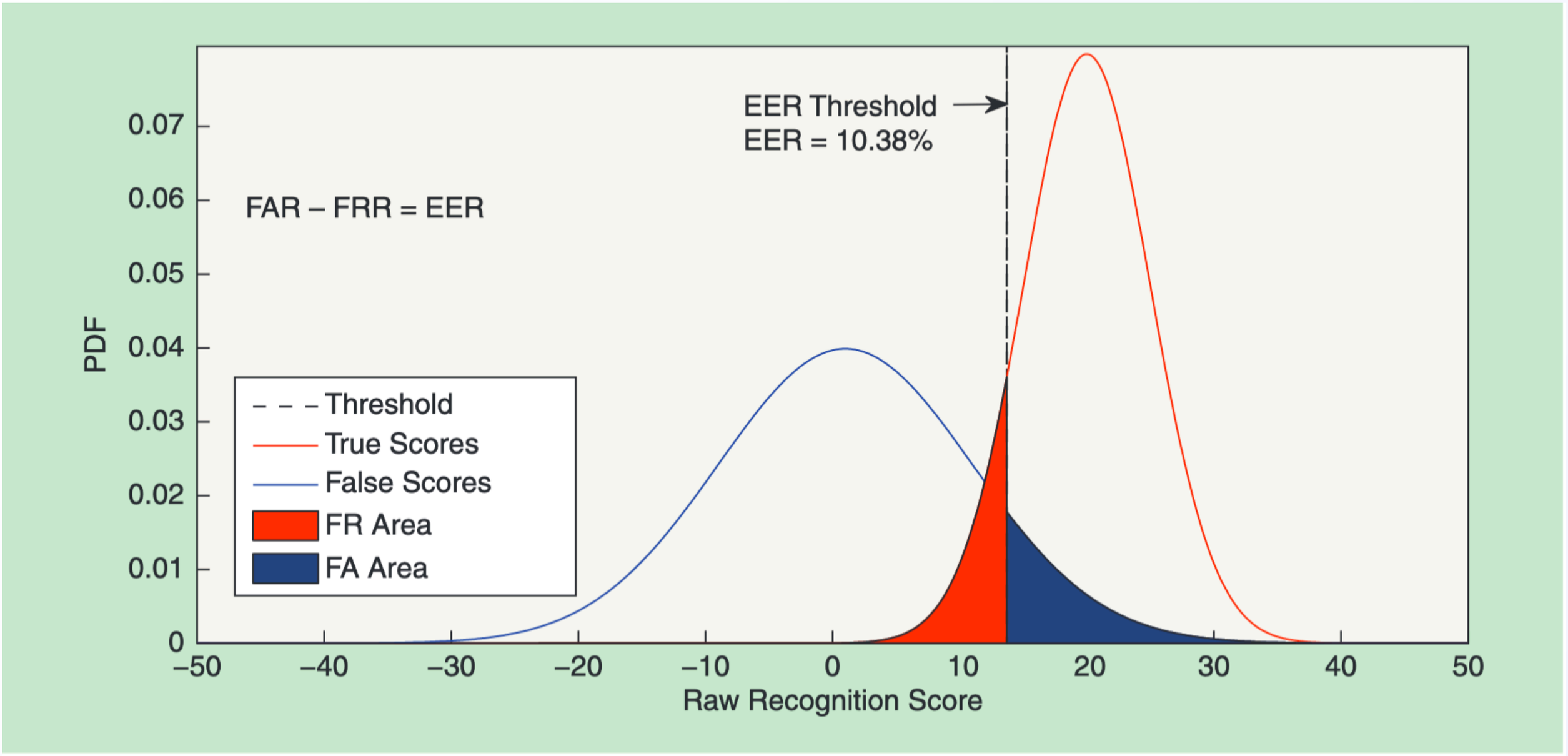
- Two types of error
  - False acceptance – grant access to an imposter: False Acceptance Rate (FAR)
  - False reject – refuse access to a genuine speaker: False Rejection Rate (FRR)

$$FAR = FalseAlarmProbability = \frac{\text{Number of imposters accepted}}{\text{Number of imposter attempts}}$$

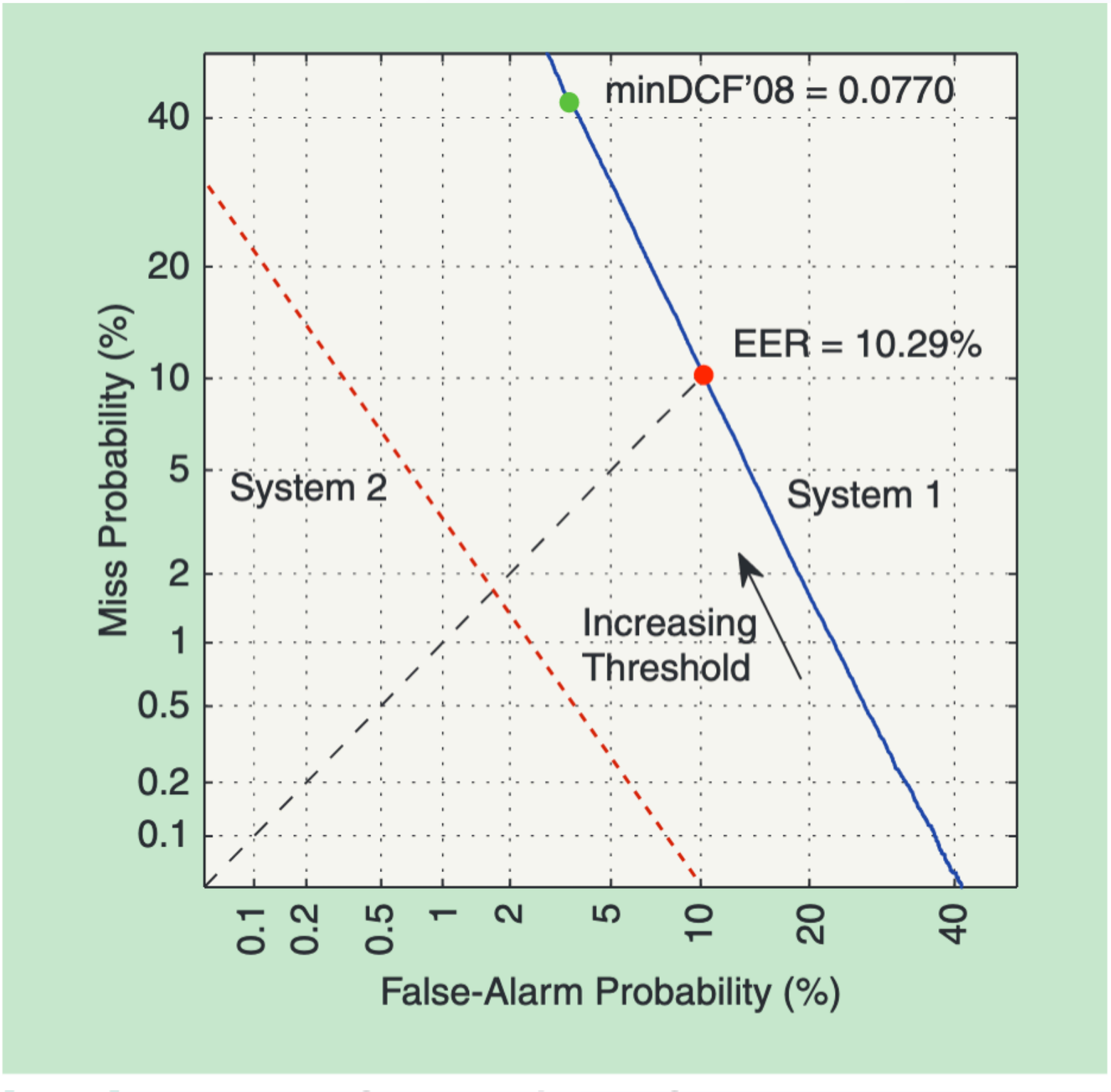
$$FRR = MissProbability = \frac{\text{Number of legitimate speakers rejected}}{\text{Number of legitimate attempts}}$$

- Control the levels of these errors by setting decision threshold
- Equal error rate – FAR and FRR values when they are equal
- DET (detection error tradeoff) curve – plots FRR (miss probability) against FAR (false alarm probability)

FA/FR



DET plots



- Scale of axes CDF based
- Assumption of normal distribution of scores

Detection cost function

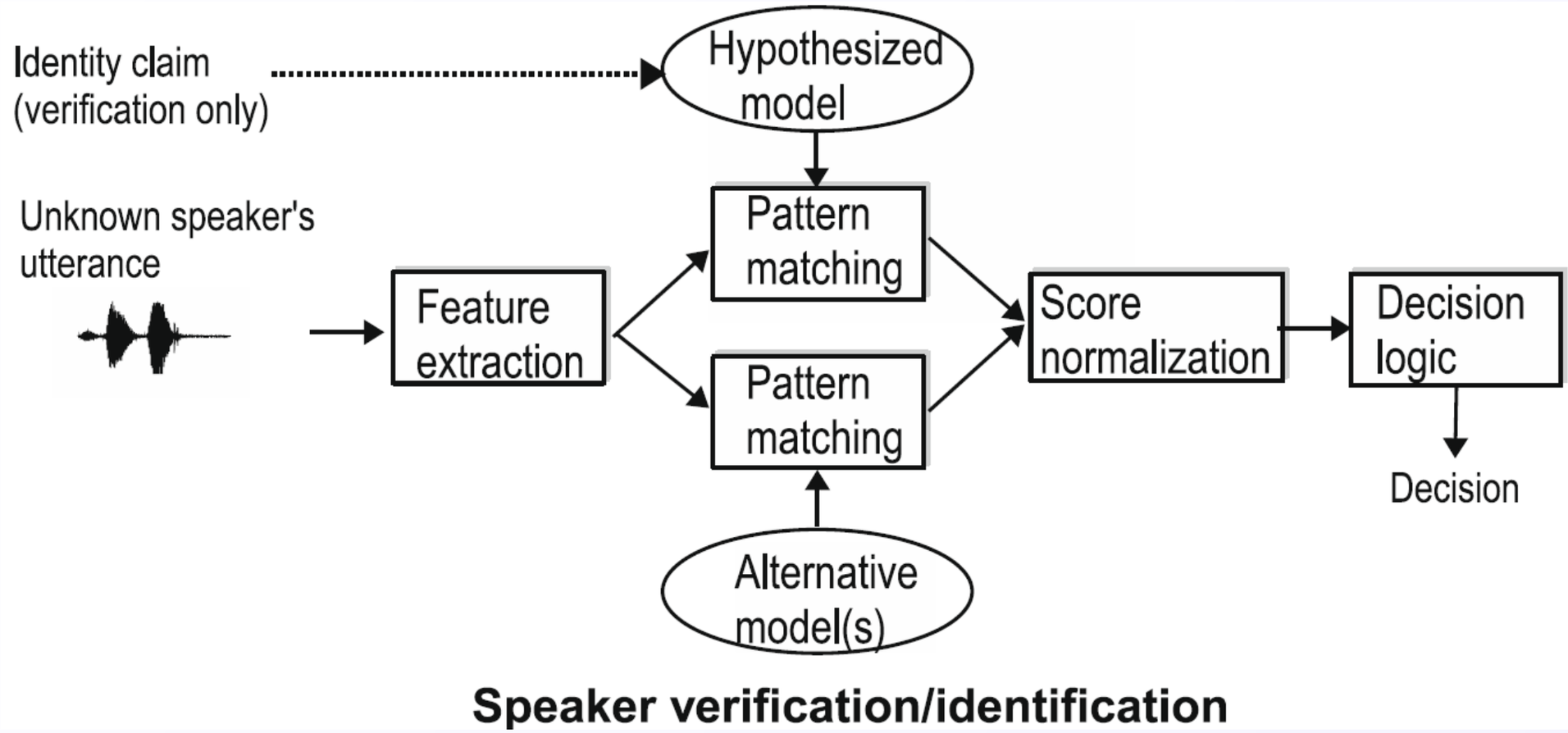
- Detection cost function takes into account
  - Cost of miss (C_{miss}) and false alarm (C_{FA}) errors
  - Prior probability of target speaker – P_{target}
  - Miss probability at threshold τ - $P_{miss}(\tau)$
  - FA probability at threshold τ - $P_{FA}(\tau)$

$$DCF(\tau) = C_{miss}P_{miss}(\tau)P_{target} + C_{FA}P_{FA}(\tau)(1 - P_{target})$$

- Set $C_{miss} > C_{FA}$ if it is better to have false alarms than it is to miss the target speaker (e.g. law enforcement applicationa)

Speaker recognition - Types

- *text dependent* vs *text independent*
- *Speaker verification vs speaker identification*

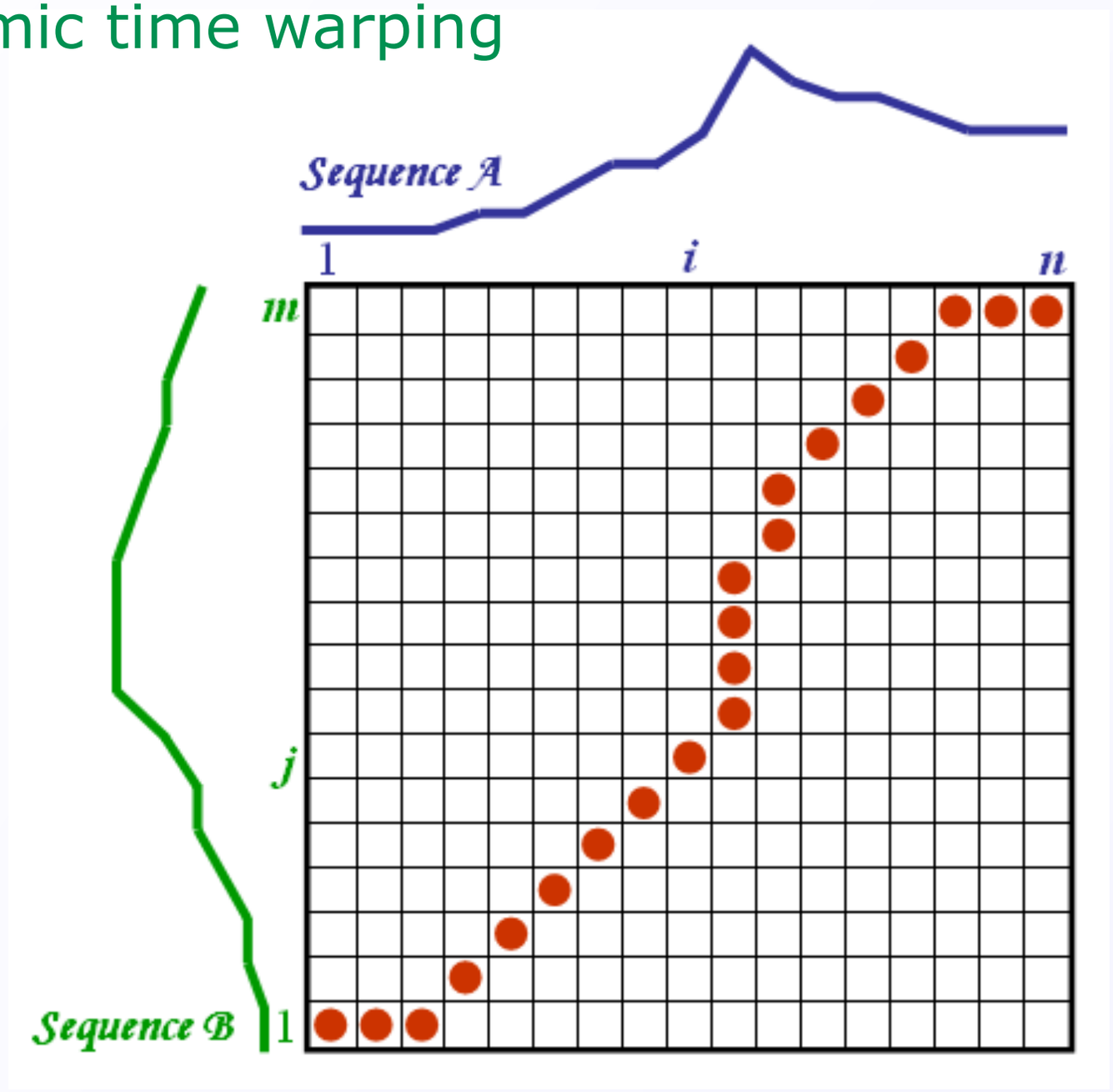


Speaker verification/identification

[Kinnunen and Li 2010]

Template models for speaker recognition

- Compute the distance between test and reference vectors
- Dynamic time warping

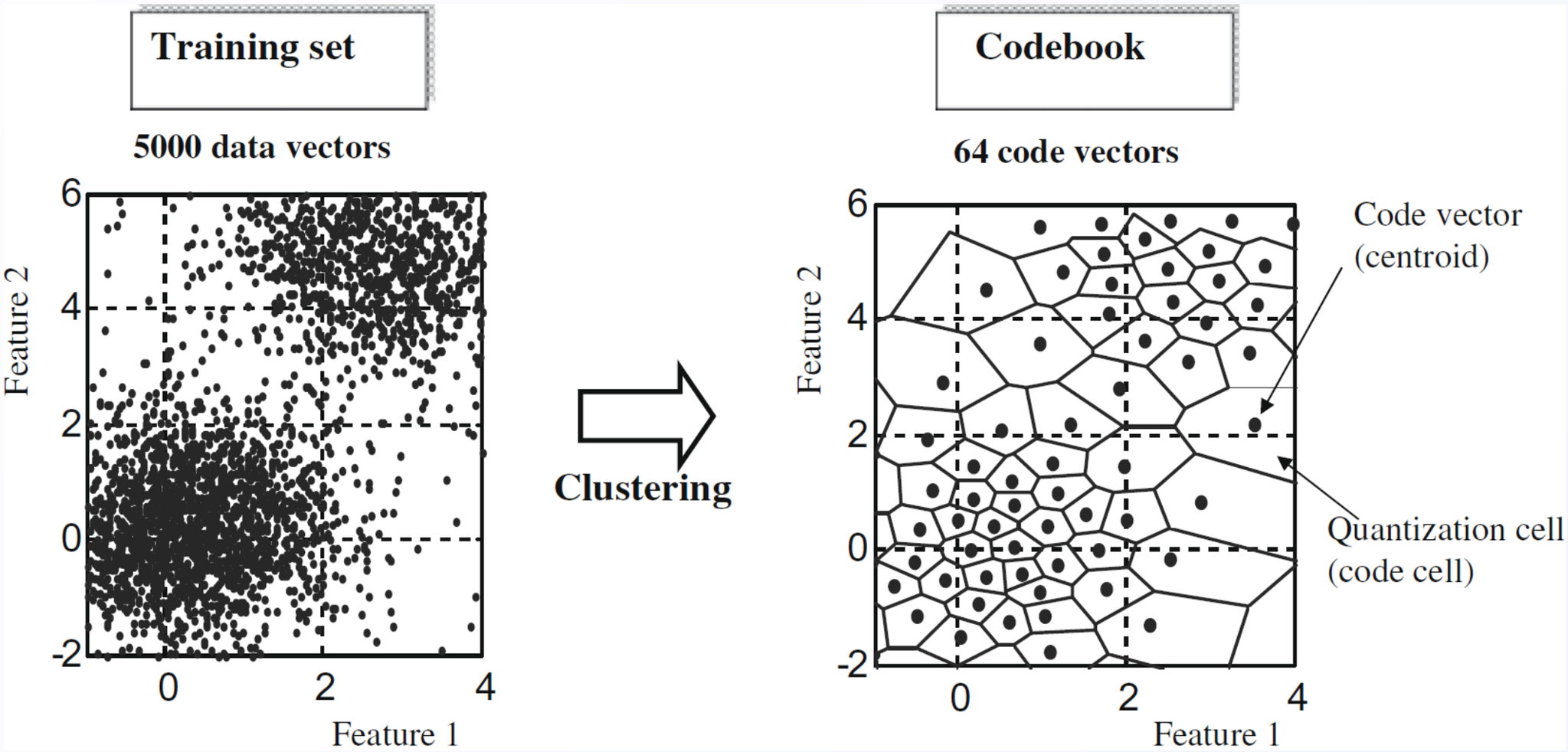


[i.stack.imgur.com/AnO31.gif Furui 1981]

Template models for speaker recognition

- Vector quantisation

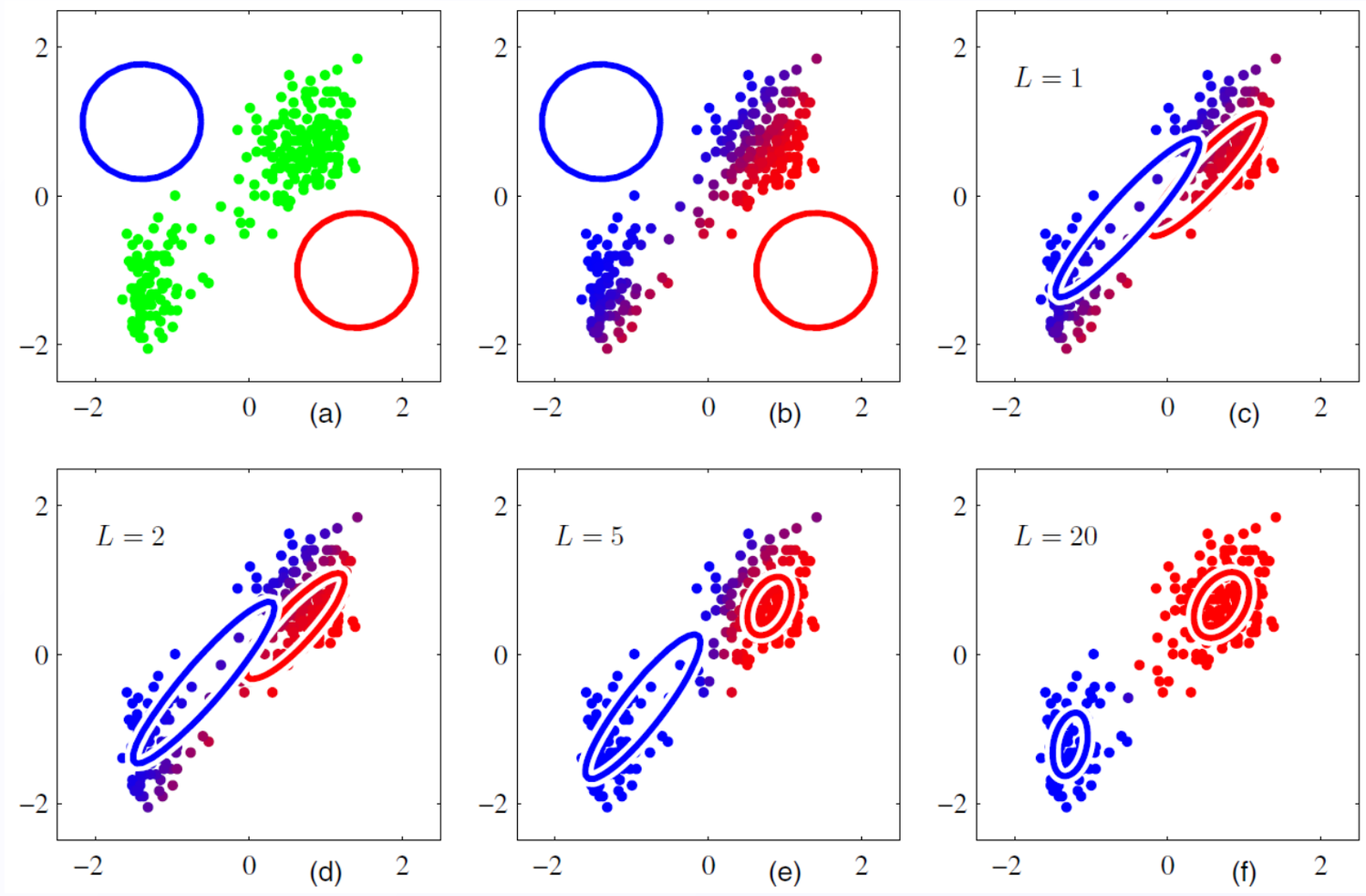
$$D(\boldsymbol{X}, \boldsymbol{R}) = \frac{1}{T} \sum_{t=1}^{T} \min_{1 \leq k \leq K} d(x_t, r_k)$$



[Kinnunen and Li 2010]

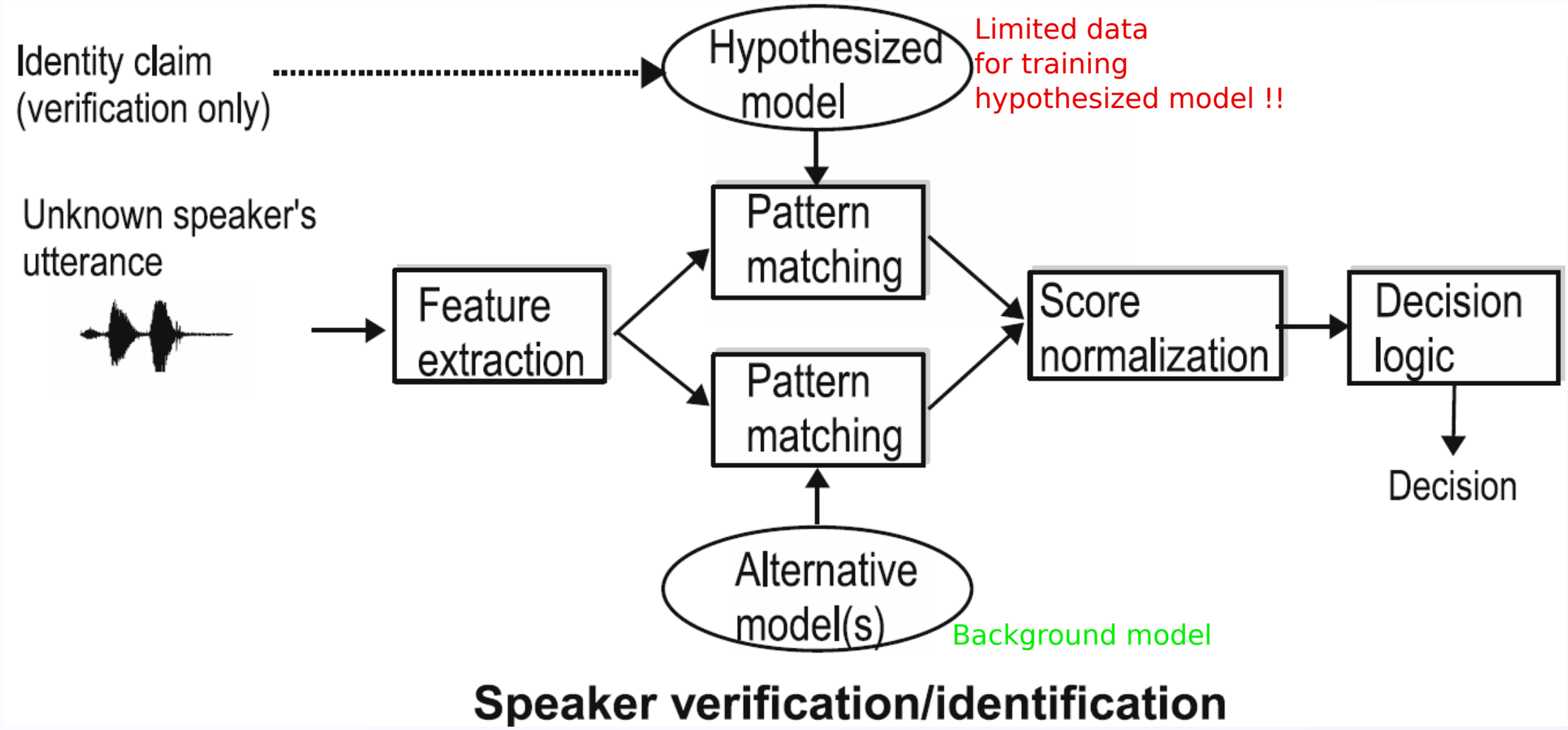
Stochastic models for speaker recognition

- Recognition decision based on accumulated scores over the frame sequence
- Centroid model to represent target speakers in the acoustic feature space
- Stochastic models : Gaussian mixture model



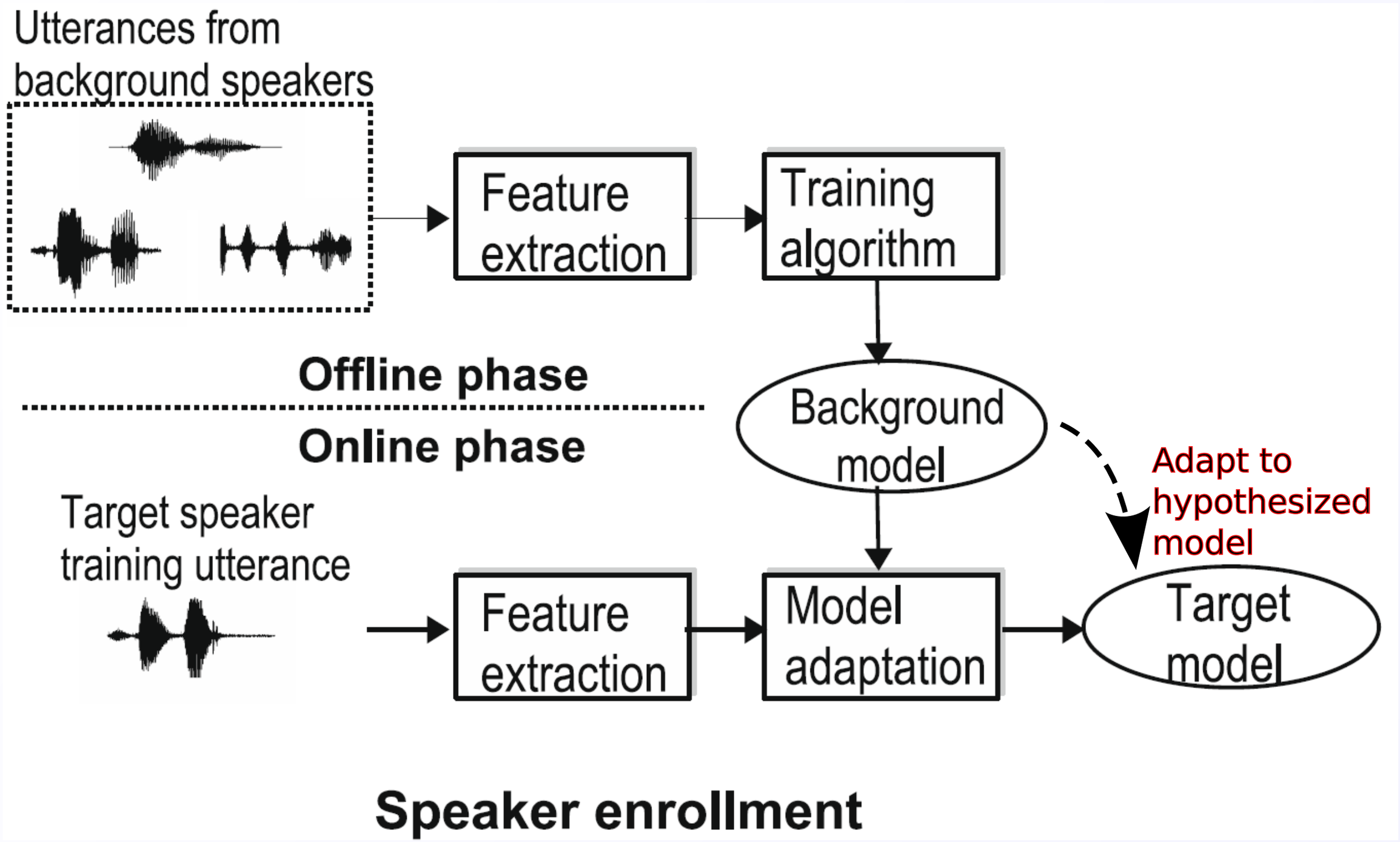
$$\mathcal{L}(\boldsymbol{X};\beta,\mu,\Sigma) = \prod_{t=1}^{T}\sum_{c=1}^{C} w_c \mathcal{N}(x_t;\mu_c,\Sigma_c)$$

Text independent speaker recognition



[Kinnunen and Li 2010]

Model enrolment



[Kinnunen and Li 2010]

Stochastic model: parameters

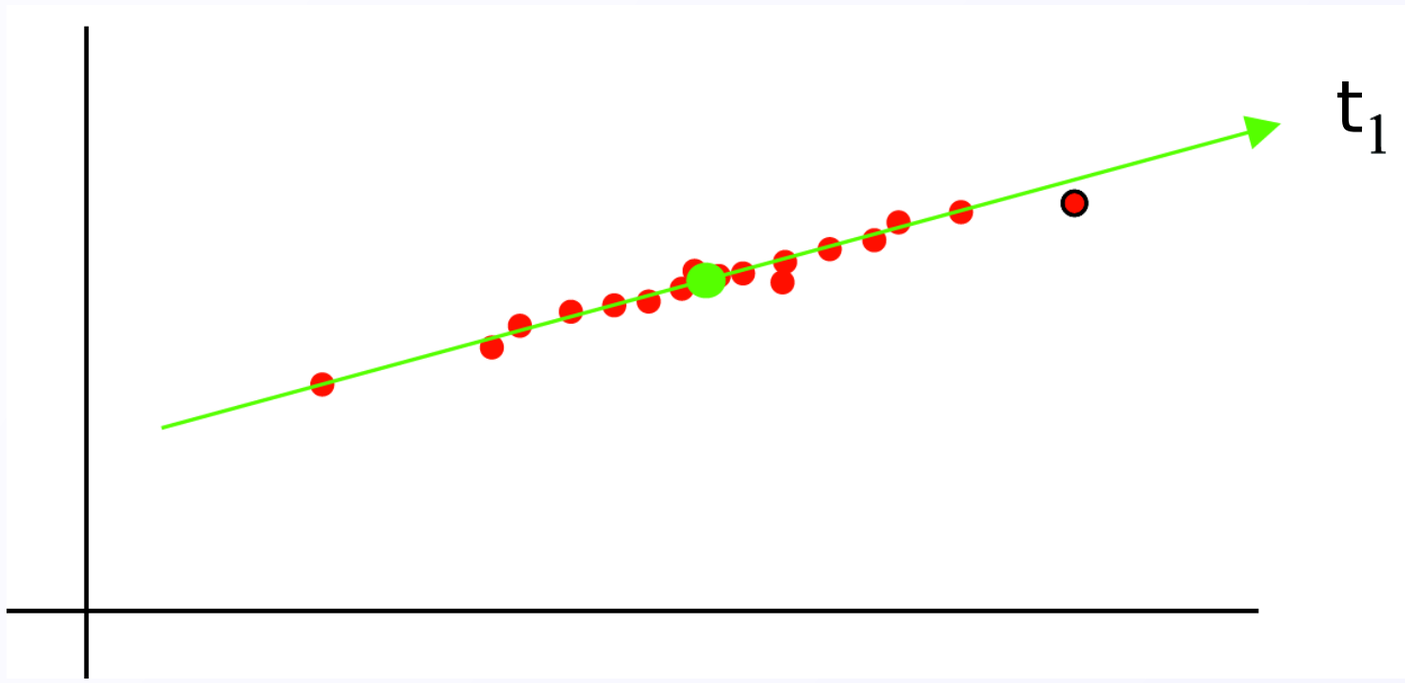
- Previous example:
  - Number of Gaussian mixture components: 2
  - Number of feature dimensions: 2

- Practical application:
  - Number of Gaussian mixture components: >1000
  - Number of feature dimensions: 39 (bottleneck: ≈100, spliced feature: ≈400)

Dimension reduction: useful technique to avoid sparsity

Subspace modelling

- Discover low-dimensional subspace in which data lies
- Exploit lower-dimensionality to allow efficient modelling



Direction of main variation

Principal component analysis

- sample statistics

$$\text{mean} \quad \bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

$$\text{covariance} \quad \mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T$$

- projection onto fixed dimension t

$$\text{mean} \quad t^T\bar{x}$$

$$\text{covariance} \quad \frac{1}{N}\sum_{n=1}^{N}(t^T x_n - t^T\bar{x})(t^T x_n - t^T\bar{x})^T = t^T\mathbf{S}t$$

PCA (2)

- Project data in the direction of t such that (transformed) covariance is maximum
- Constrain *t* to be a unit vector

$$\arg\max_t L = \arg\max_t t^T \mathbf{S} t + \lambda(1 - t^T t)$$

$$\text{Set } \frac{\partial L}{\partial t} = 0$$

$$\mathbf{S}t = \lambda t \quad (t^T \mathbf{S} t = t^T \lambda t = \lambda)$$

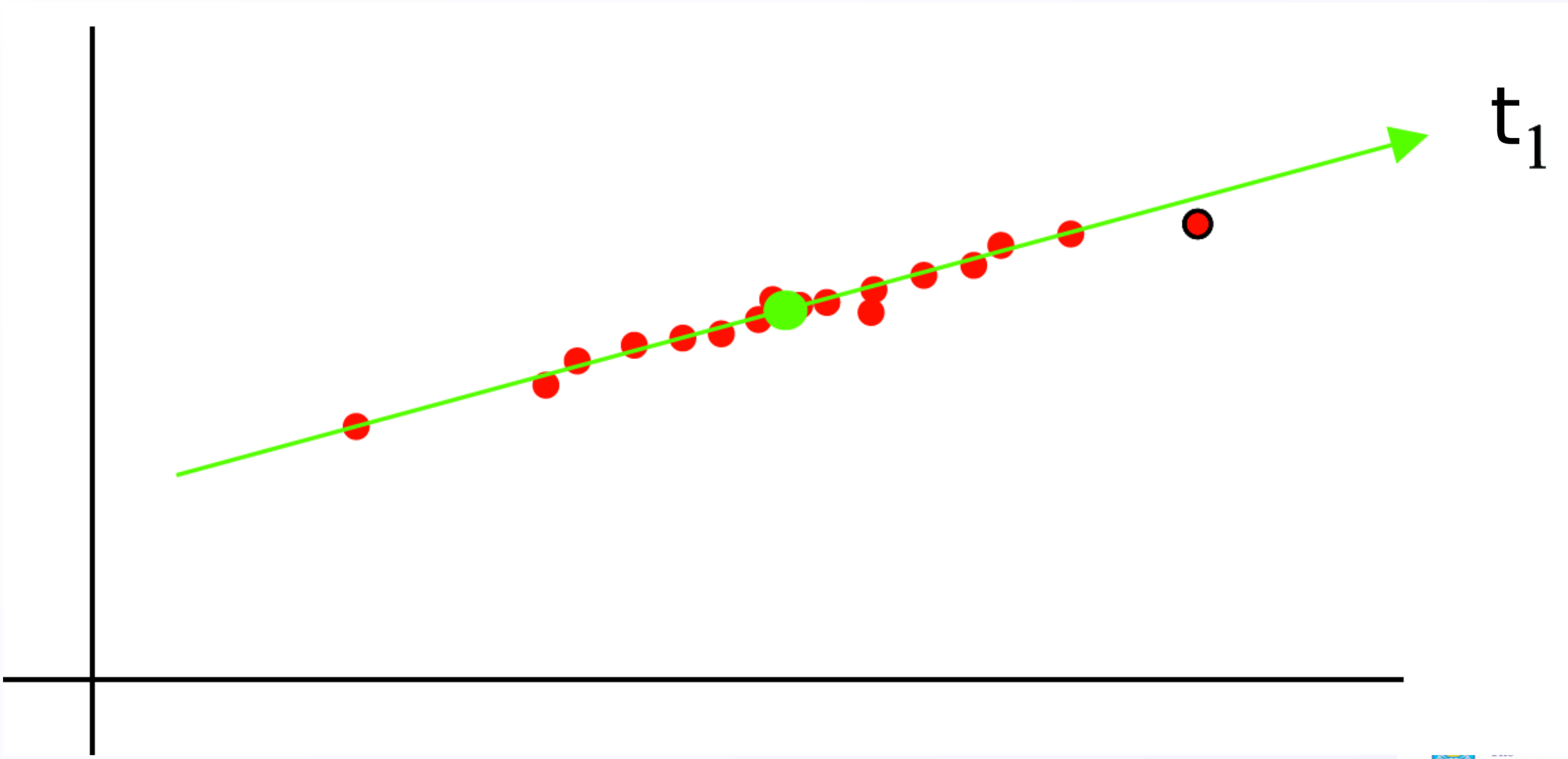
- *t* is an eigenvector of ***S***
- λ is the eigenvalue or the projected variance

How is this done in practice ?

- Eigenvectors $\rightarrow$ basis vectors; Set of basis vectors $\rightarrow$ vector space
- $\mathbf{T} = [t_1, t_2, t_3, \ldots t_M]$ is a M-dimensional reduced representation of the observed data x
- M = Number of basis vectors = Dimension of the vector space
- Example: MFCC
  - Number of dimension $F = 39$ $(x \in \mathbb{R}^{39})$
  - M basis vectors $[t_1 \ldots t_M] \in \mathbb{R}^{39}$ (ranked by eigenvalues)
  - Form a new vector space $\mathbb{M}$

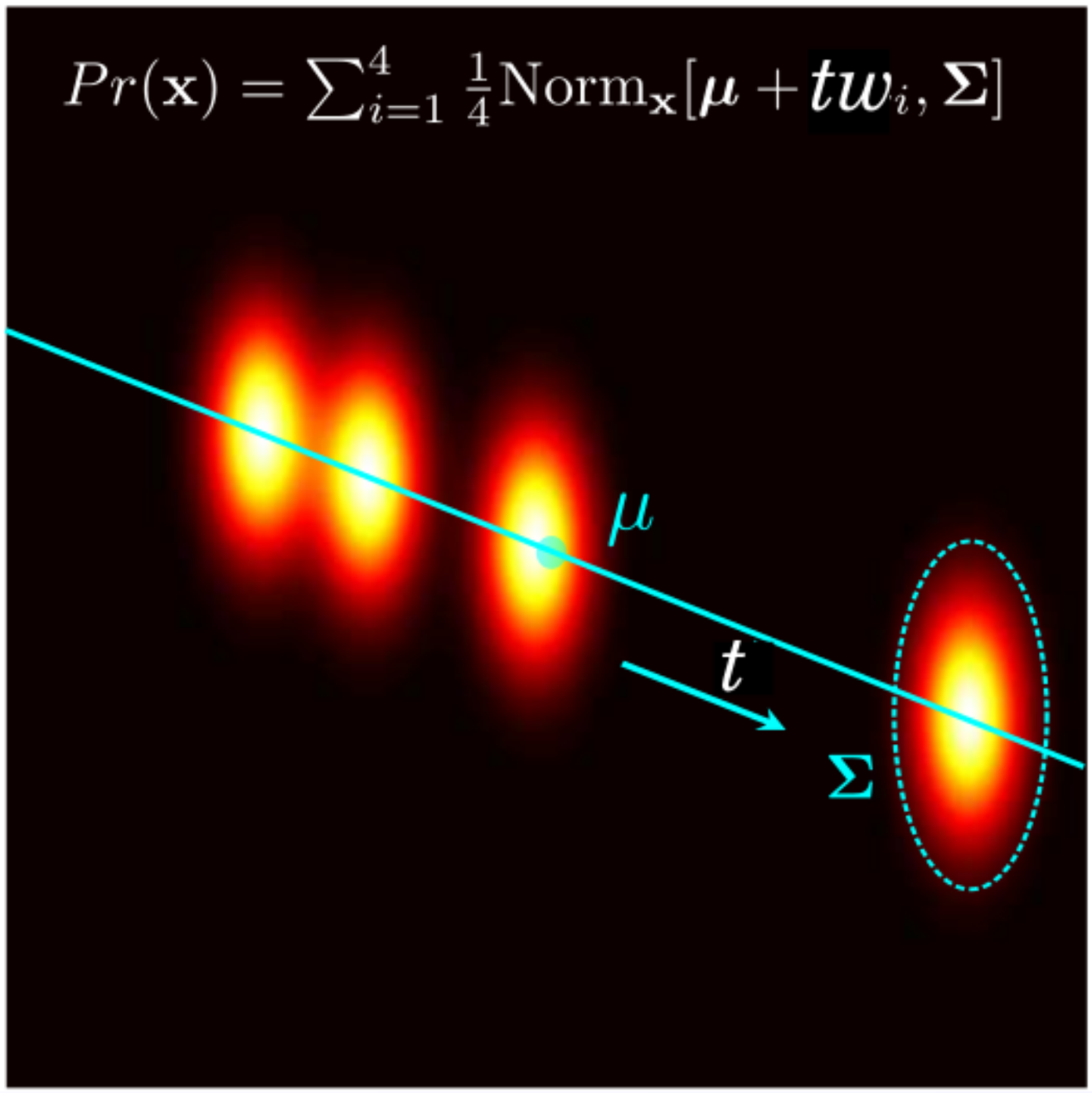
Probabilistic view

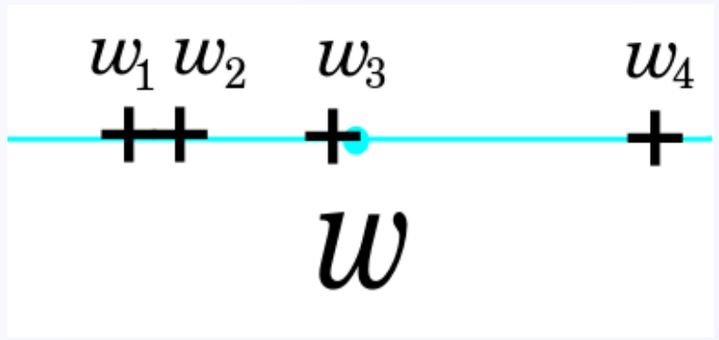
- A generative modelling view
- Extension to a mixture model [Tipping and Bishop 1999]
- $\mathbf{T} = [t_1 \ldots t_M]$ are no longer the leading M eigenvectors of $\mathbf{S}$



Mixtures of Gaussians

- View *x* as generated by a mixture of Gaussians with mean tw_i

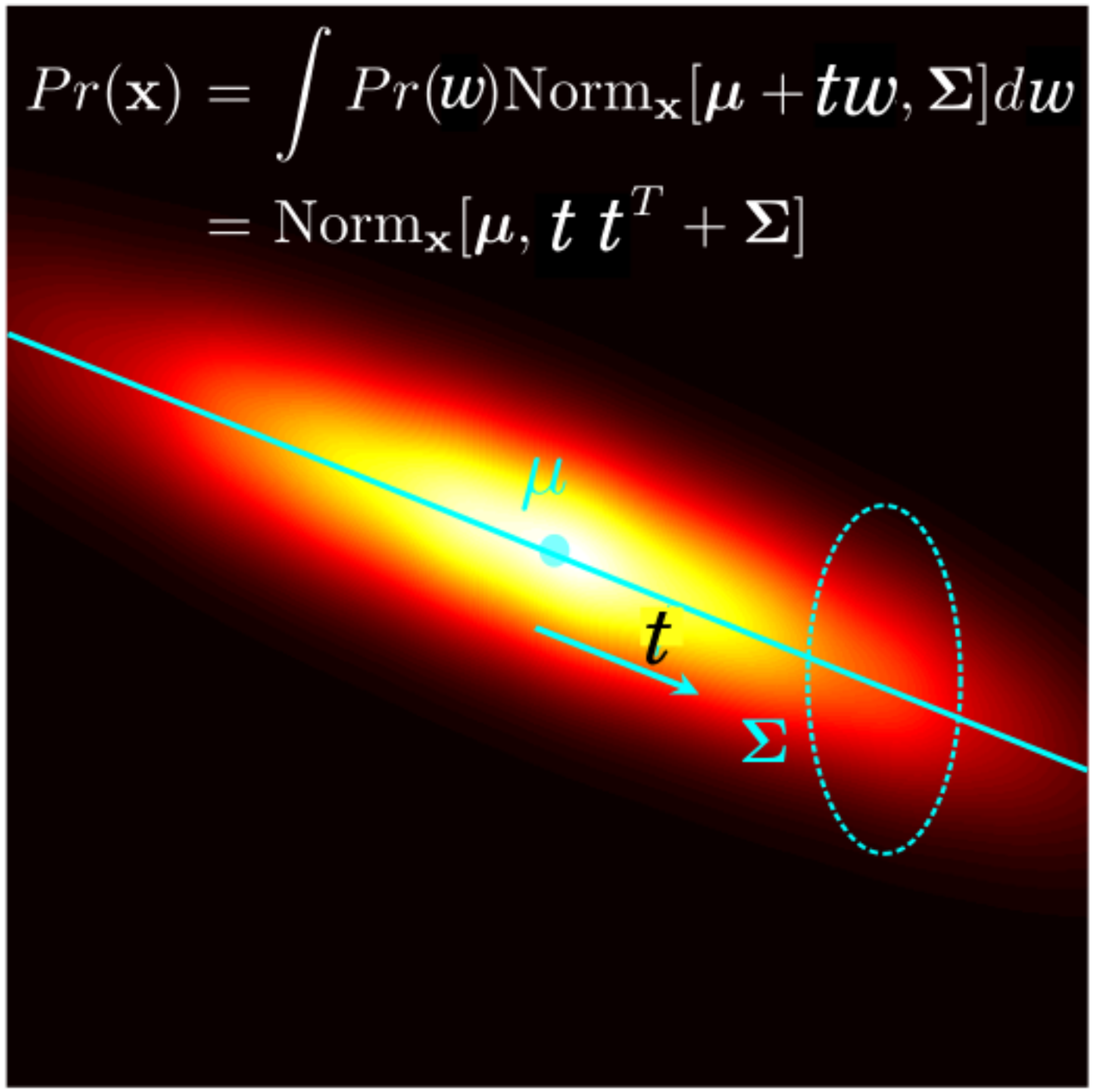


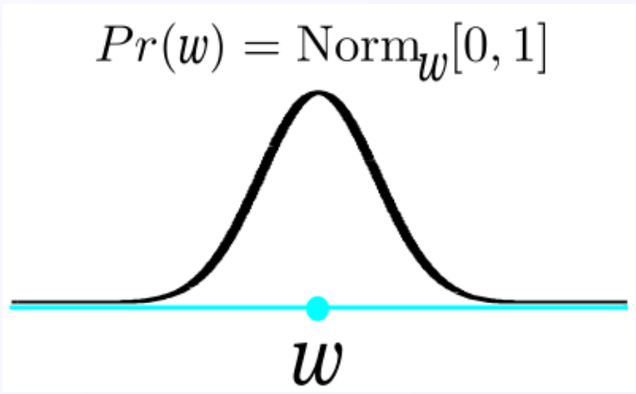


[Prince 2012]

Probabilistic PCA

- Extending to continous prior distribution P(w)





[Prince 2012]

Supervector systems for speaker recognition

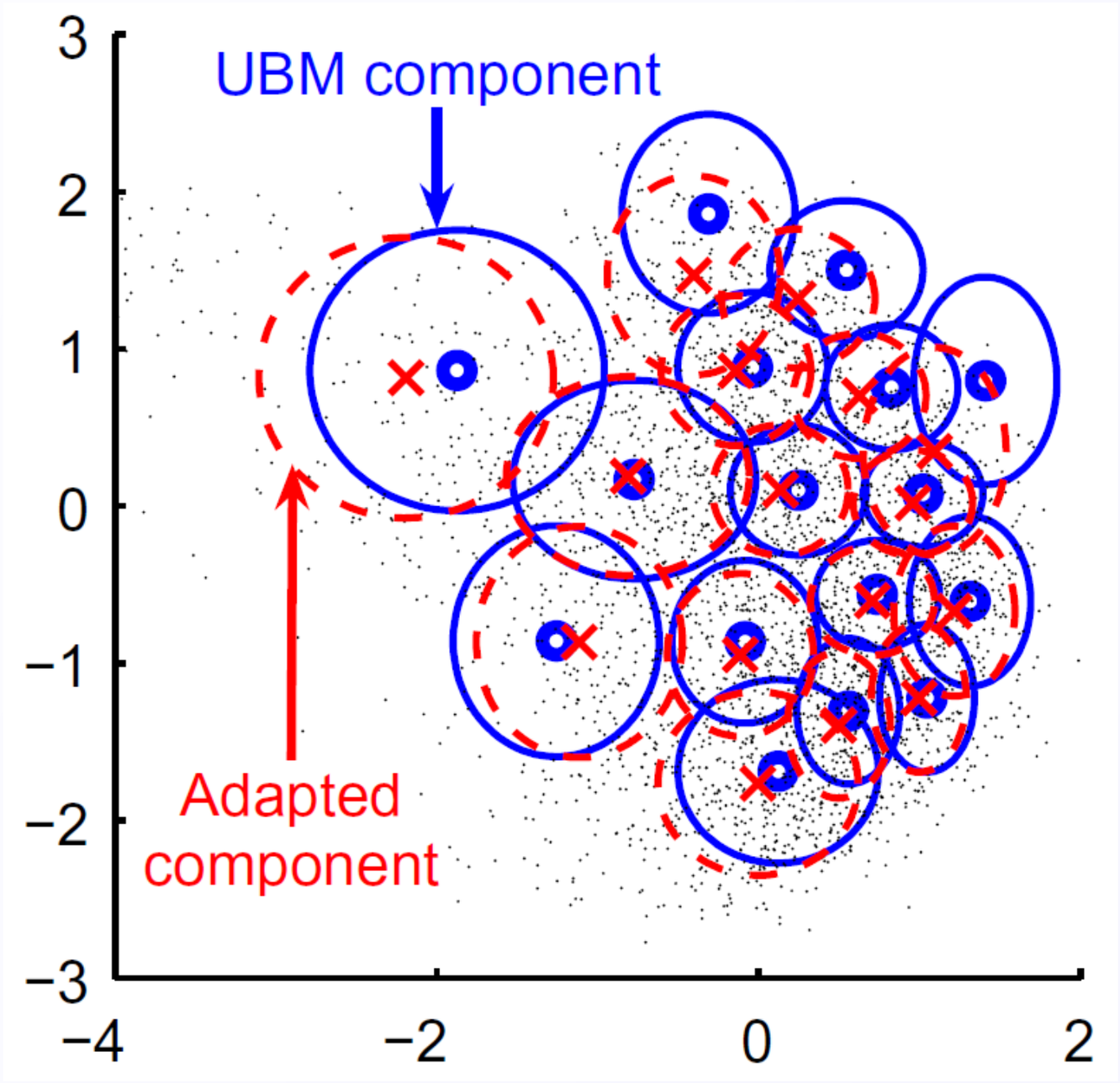
- In speaker recognition runtime, an unknown voice is represented by some feature vector for the comparison with speaker models

- Supervector is a concatenation of multiple vectors, typically the mean vector of all Gaussian components

- Dimension of supervector =
  C (Number of Gaussians) × F (Feature dimension)

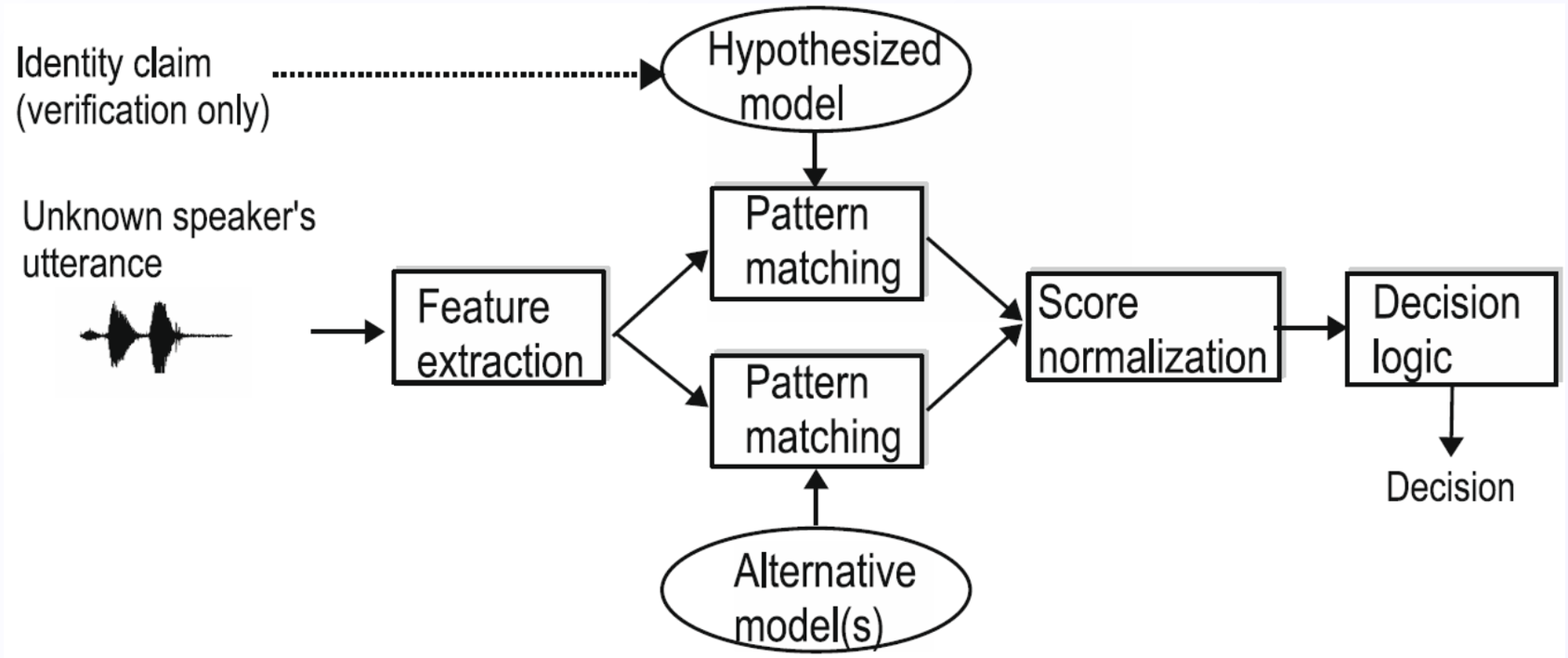
- Fixed length representation for a sentence

Visualising the feature space



- [Kinnunen and Li 2010]
- Recognition: Likelihood ratio between true and imposter model

Speaker recognition



[Kinnunen and Li 2010] **Speaker verification/identification**

- Hypothesized model and alternative models are derived from background model
- Instead of training target and imposter models, discover the latent variable corresponding to the utterance
- Compare the latent varibles between target and test sentence

Gaussian supervector

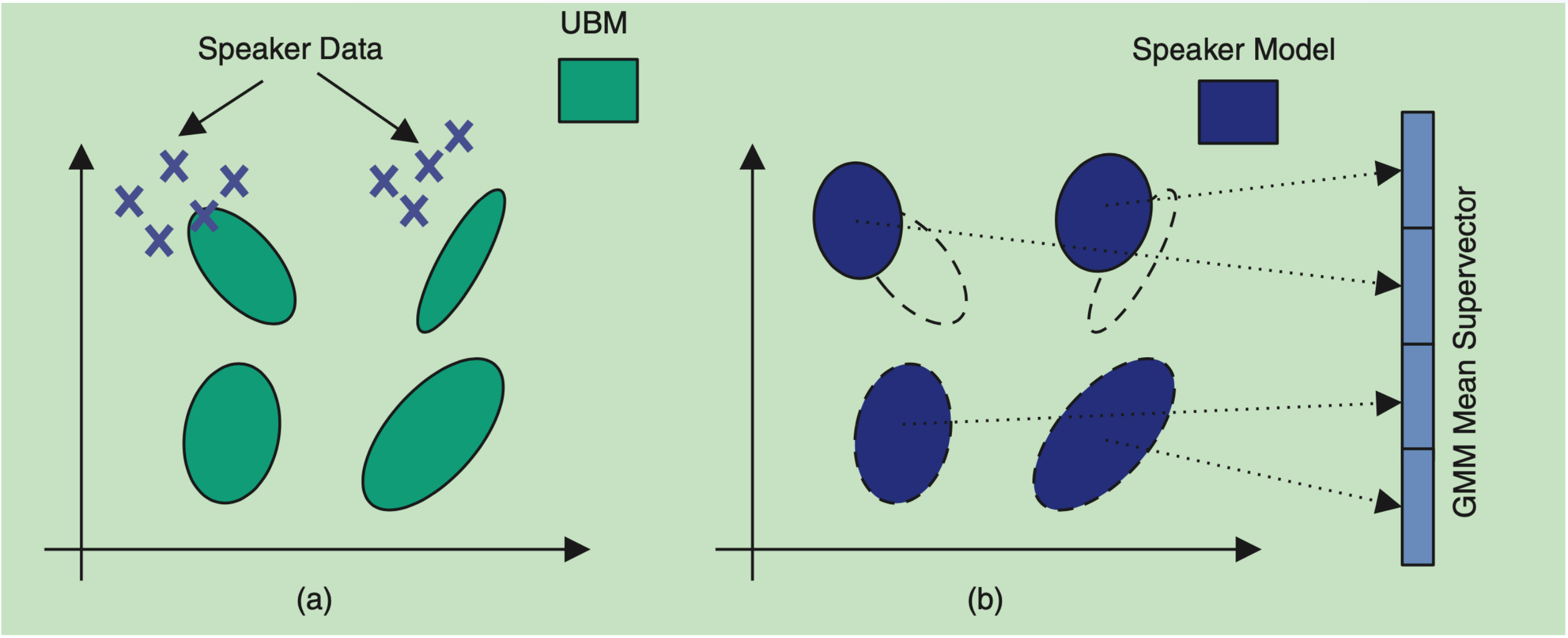
- For a sentence $\boldsymbol{x} = [x_1 \dots x_T]$, the overall likelihood is,

$$\mathcal{L}(\boldsymbol{X}; \beta, \mu, \Sigma) = \prod_{t=1}^{T} \sum_{c=1}^{C} \beta_c \mathcal{N}(x_t; \mu_c, \Sigma_c)$$

- Find optimal statistical model parameters μ to represent the data

- With two exemplars "True", "Imposter" and a test sentence "Test"
  - Raw feature scoring
  - Scoring with Gaussian supervector distance: $\text{Dist}(\mu_{\text{Test}}, \mu_{\text{True}})$ and $\text{Dist}(\mu_{\text{Test}}, \mu_{\text{Imposter}})$

Supervectors illustrated



Sufficient statistics for general learning

- Find μ to represent sequence $\boldsymbol{X}$

$$\mathcal{L}(\boldsymbol{X};\beta,\mu,\Sigma) = \prod_{t=1}^{T}\sum_{c=1}^{C}\beta_c\mathcal{N}(x_t;\mu_c,\Sigma_c)$$

$$\log\mathcal{L}(\boldsymbol{X};\beta,\mu,\Sigma) = \log\prod_{t=1}^{T}\sum_{c=1}^{C}\beta_c\ldots = \sum_{t=1}^{T}\log\sum_{c=1}^{C}\beta_c\ldots$$

$$= \sum_{t=1}^{T}\sum_{c=1}^{C}\gamma_t^c\log\mathcal{N}(x_t;\mu_c,\Sigma_c) + \text{const}$$

$$\gamma_t^c = \frac{\mathcal{N}(x_t;\mu_c,\Sigma_c)}{\sum_{k=1}^{C}\mathcal{N}(x_t;\mu_k,\Sigma_k)}$$

Sufficient statistics

$$N_c = \sum_{t=1}^{T} \gamma_t^c$$

$$f_c = \sum_{t=1}^{T} \gamma_t^c x_t$$

$$S_c = \sum_{t=1}^{T} \gamma_t^c x_t x_t^T$$

$$\mathbb{F} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_C \end{bmatrix}$$

$$\mathbb{N} = \begin{bmatrix} N_1\mathbf{I} & 0 & \dots & 0 \\ 0 & N_2\mathbf{I} & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & N_C\mathbf{I} \end{bmatrix}$$

$$\mathbb{S} = \begin{bmatrix} S_1 & 0 & \dots & 0 \\ 0 & S_2 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & S_C \end{bmatrix}$$

Inferred Likelihood ..

$$\log \mathcal{L} = \sum_{t=1}^{T}\sum_{c=1}^{C} \gamma_t^c \log \mathcal{N}(x_t; \mu_c, \Sigma_c) + \text{const}$$

$$= -\frac{1}{2}\sum_{c=1}^{C}\sum_{t=1}^{T} \gamma_t^c x_t^T \Sigma_c^{-1} x_t + \frac{1}{2}\sum_{c=1}^{C}\sum_{t=1}^{T} \gamma_t^c \left(x_t^T \Sigma_c^{-1}\mu_c + \mu_c^T \Sigma_c^{-1} x_t\right)$$

$$- \frac{1}{2}\sum_{c=1}^{C}\sum_{t=1}^{T} \gamma_t^c \mu_c^T \Sigma_c^{-1}\mu_c + \text{const}$$

$$= \ldots$$

$$= -\frac{1}{2} Tr(\mathbb{S}\Sigma^{-1}) + \mu^T \Sigma^{-1}\mathbb{F} - \frac{1}{2}\mu^T \mathbb{N}\Sigma^{-1}\mu + \text{const}$$

- Optimising μ for maximum $\mathcal{L}$

$$\frac{\partial \log \mathcal{L}}{\partial \mu} = \frac{\partial}{\partial \mu}\left(\mu^T \Sigma^{-1}\mathbb{F} - \frac{1}{2}\mu^T \mathbb{N}\Sigma^{-1}\mu\right)$$

- High dimensionality of μ (C × F) and sparse data X make estimation difficult

I-Vectors

- Recall the likelihood calculation of data

$$\log \mathcal{L}(\boldsymbol{X}; \mu) = \mu^T \Sigma^{-1} \mathbb{F} - \frac{1}{2} \mu^T \mathbb{N} \Sigma^{-1} \mu + \text{const}$$

- For high dimensional μ_c, use factor analysis

$$\mu_c = m + T w_c$$

- The optimal posterior estimate w is the i-vector of the sentence

Likelihood and posterior

- Assume $m = 0$, substituting $\mu = Tw$, assume T is known,

$$\log \mathcal{L}(\boldsymbol{X}; w) = \log p(\boldsymbol{X}|w) = w^T T^T \Sigma^{-1} \mathbb{F} - \frac{1}{2} w^T T^T \mathbb{N} \Sigma^{-1} T w + \text{const}$$

- Consider prior $p(w) \sim \mathcal{N}(w; 0, \mathbb{I})$,

$$\log p(w) = -\frac{1}{2} w^T w + \text{const}$$

- Find w which gives the maximum posterior probability

$$\begin{aligned}
\log p(w|\boldsymbol{X}) &\propto \log p(\boldsymbol{X}, w) \\
&= \log p(\boldsymbol{X}|w) + \log p(w) \\
&= w^T T^T \Sigma^{-1} \mathbb{F} - \frac{1}{2} w^T T^T \mathbb{N} \Sigma^{-1} T w - \frac{1}{2} w^T w + \text{const} \\
&= -\frac{1}{2} w^T (\mathbf{I} + T^T \mathbb{N} \Sigma^{-1} T) w + w^T T^T \Sigma^{-1} \mathbb{F} + \text{const}
\end{aligned}$$

Posterior probability calculation

- Define $L = \mathbf{I} + T^T \mathbb{N} \Sigma^{-1} T$

$$\log p(w|\boldsymbol{X}) \propto -\frac{1}{2} w^T (\mathbf{I} + T^T \mathbb{N} \Sigma^{-1} T) w + w^T T^T \Sigma^{-1} \mathbb{F}$$

$$= -\frac{1}{2} \left(w^T L w - 2 w^T L L^{-1} T^T \Sigma^{-1} \mathbb{F} \right)$$

$$\propto -\frac{1}{2} \left(w^T L w - 2 w^T L (L^{-1} T^T \Sigma^{-1} \mathbb{F}) \right.$$

$$\left. + (L^{-1} T^T \Sigma^{-1} \mathbb{F})^T L (L^{-1} T^T \Sigma^{-1} \mathbb{F}) \right)$$

$$= -\frac{1}{2} \left(w - (L^{-1} T^T \Sigma^{-1} \mathbb{F}) \right)^T L \left(w - (L^{-1} T^T \Sigma^{-1} \mathbb{F}) \right)$$

Posterior probability calculation

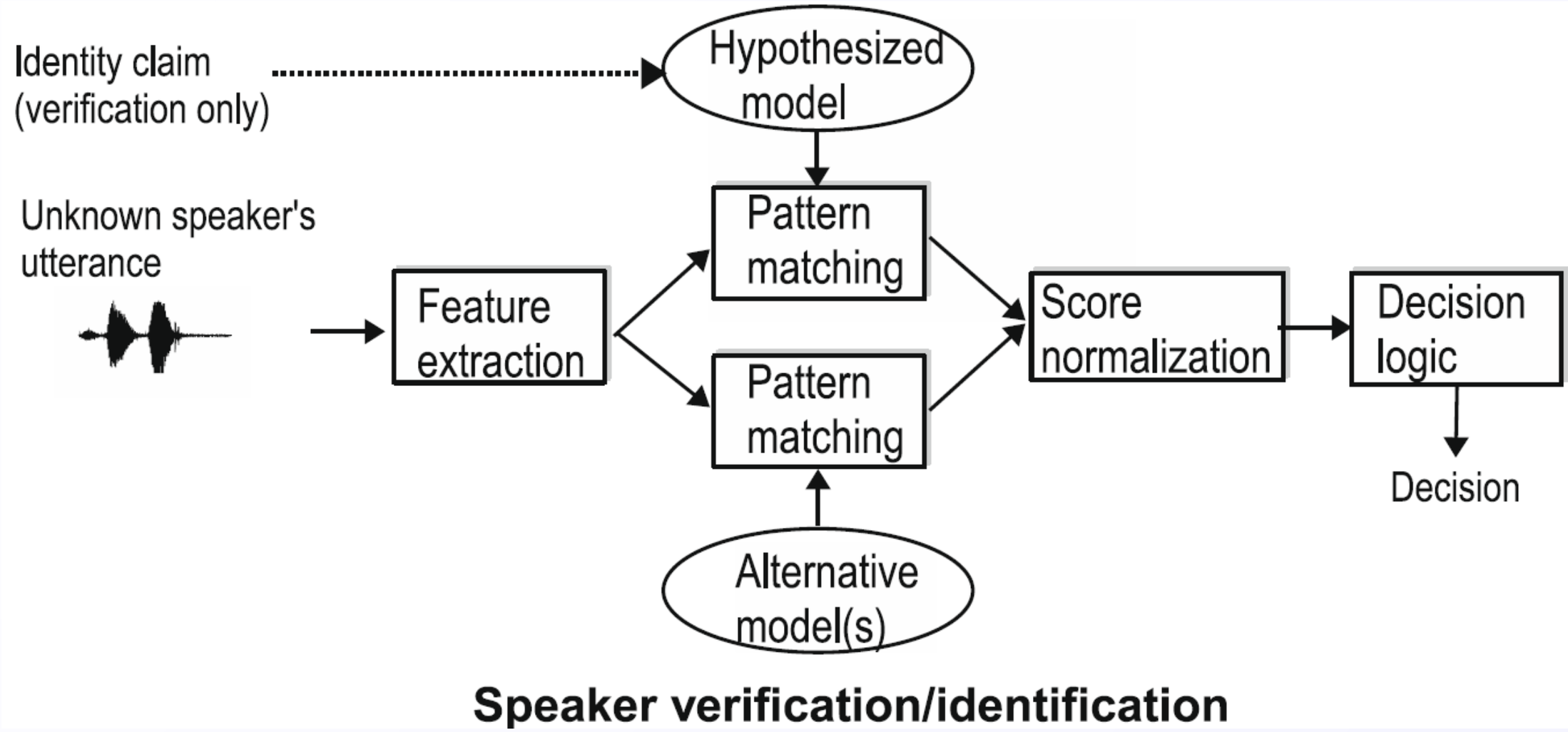
- w can be described with the Gaussian distribution equation

$$\log p(w|\boldsymbol{X}) \propto -\frac{1}{2}\left(w - (L^{-1}T^T\Sigma^{-1}\mathbb{F})\right)^T L \left(w - (L^{-1}T^T\Sigma^{-1}\mathbb{F})\right)$$
$$\sim \mathcal{N}(w; L^{-1}T^T\Sigma^{-1}\mathbb{F}, L^{-1}) \quad (1)$$

- The i-vector is the posterior mean $L^{-1}T^T\Sigma^{-1}\mathbb{F}$
- Hyperparameter T is further found by expectation maximisation

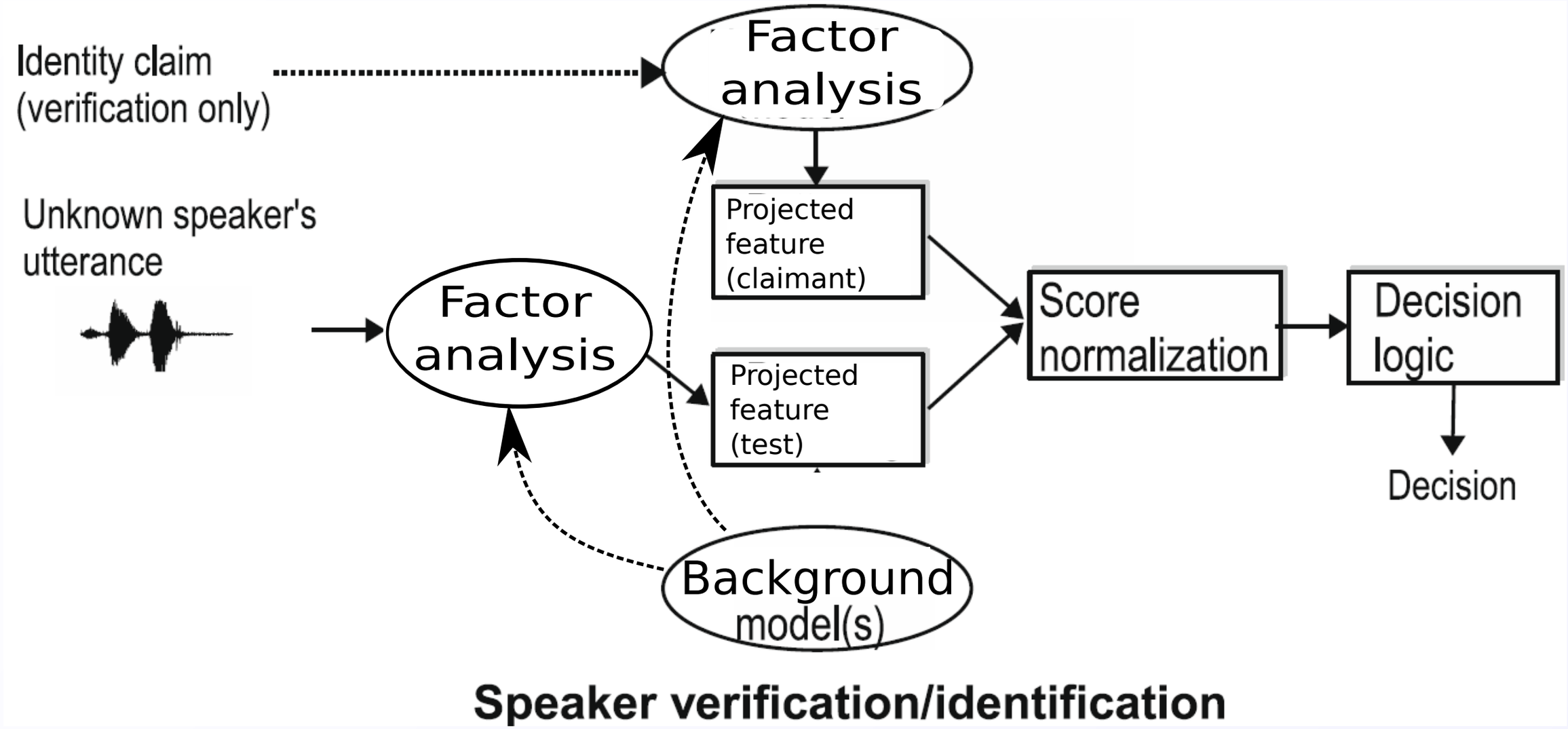
Speaker recognition systems

- Revisiting stochastic, text-independent system



Speaker verification/identification

Supervector / I-vector system



Speaker verification/identification

Score handling

- Gaussian mixture model [D. A. Reynolds, Quatieri, and Dunn 2000]
- Support vector machine [Campbell et al. 2006]
- Cosine distance [Dehak et al. 2011]
- Bayesian methods (e.g. PLDA) [Kenny et al. 2013]
- Intersession variability compensation [Dehak et al. 2011]
- Within-class covariance normalisation

$$k(w_1, w_2) = w_1^T \mathrm{Cov}^{-1} w_2$$

Score computation

- Speaker verification involves computing a score $f(w_{target}, w_{test})$ between the target and test i-vectors
- Cosine score

$$f_{cos}(w_{target}, w_{test}) = \frac{w_{target} \cdot w_{test}}{||w_{target}||||w_{test}||}$$

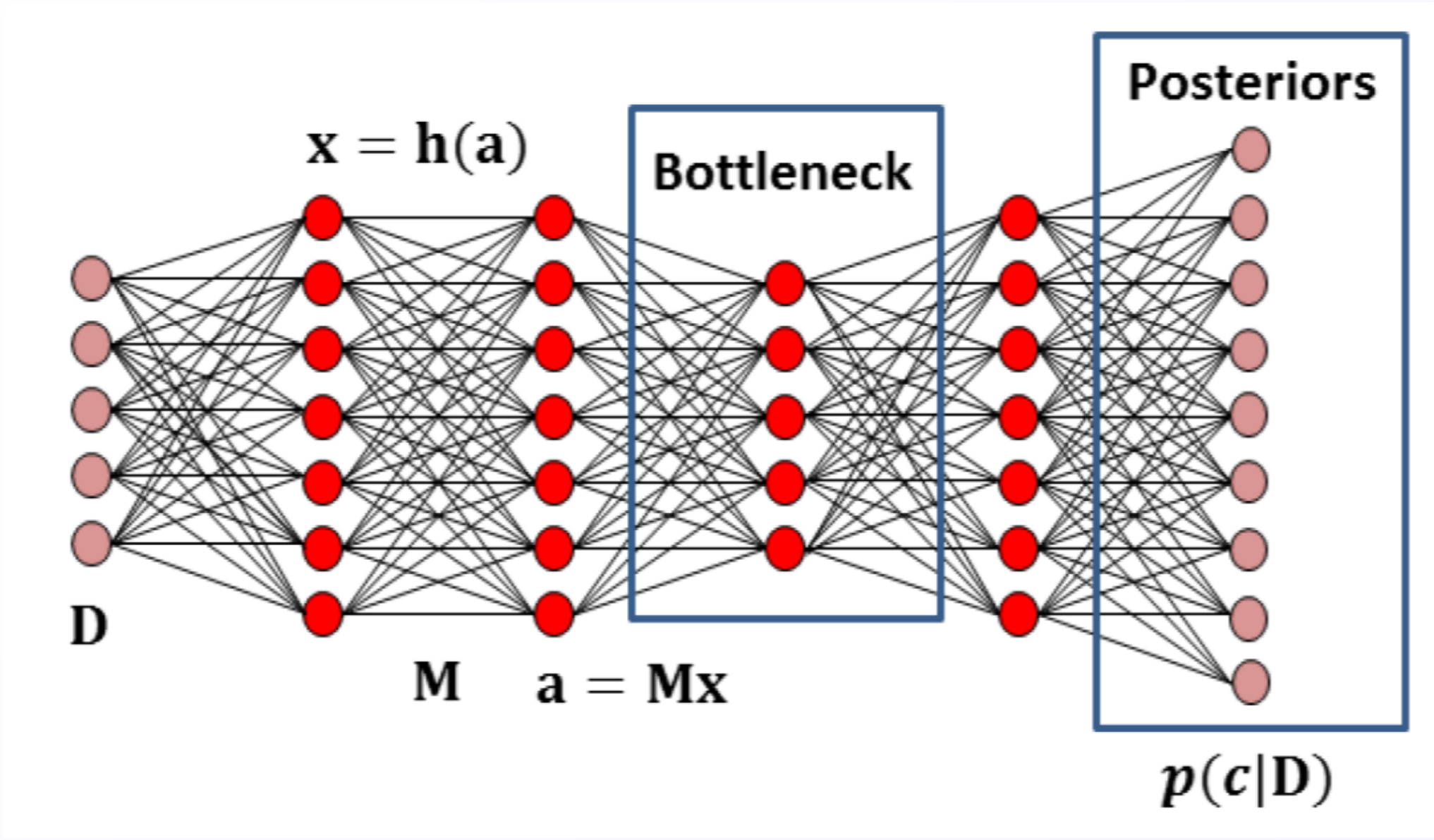
- Probabilistic linear discriminant analysis (PLDA) – probabilistic model that accounts for speaker variability and channel variability.
  Can be used to compute the log likelihood ratio, so

$$f_{plda}(w_{target}, w_{test}) = logp(w_{target}, w_{test}|H_1)logp(w_{target}|H_0)p(w_{test}|H_0)$$

  where H_1 is the hypothesis that the test and target speakers are the same, H_0 is the hypothesis they are different.
- PLDA is current-state of the art for scoring i-vectors

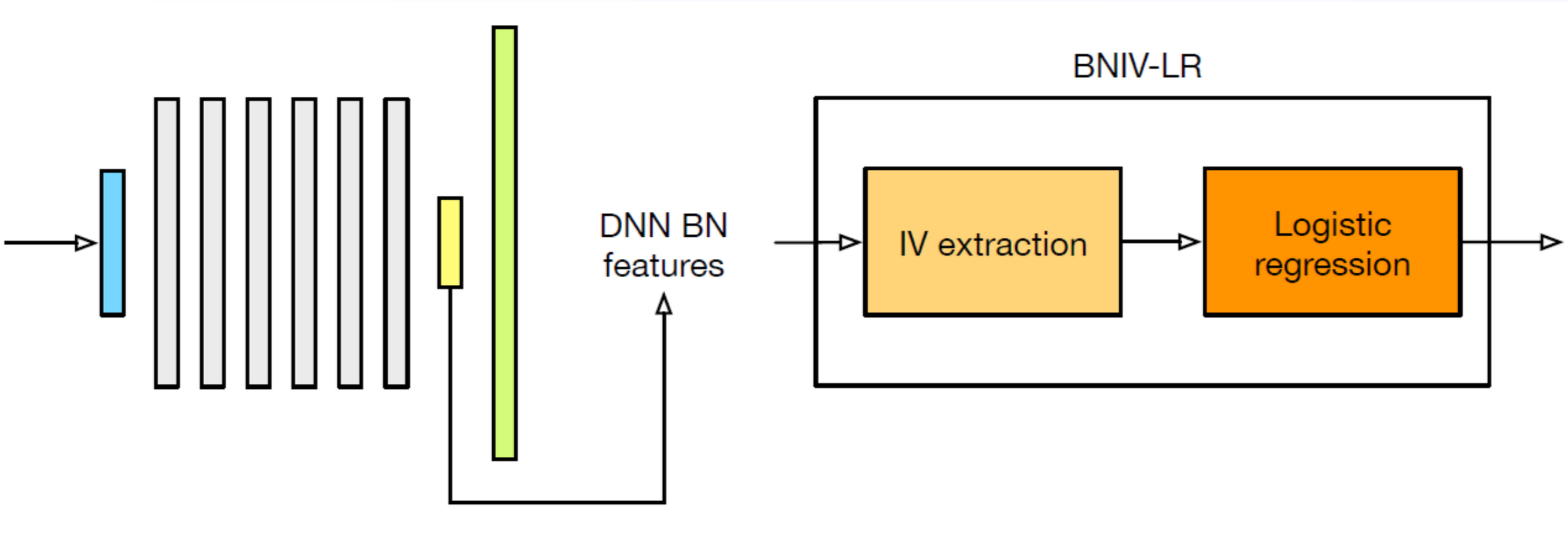
Neural network approaches

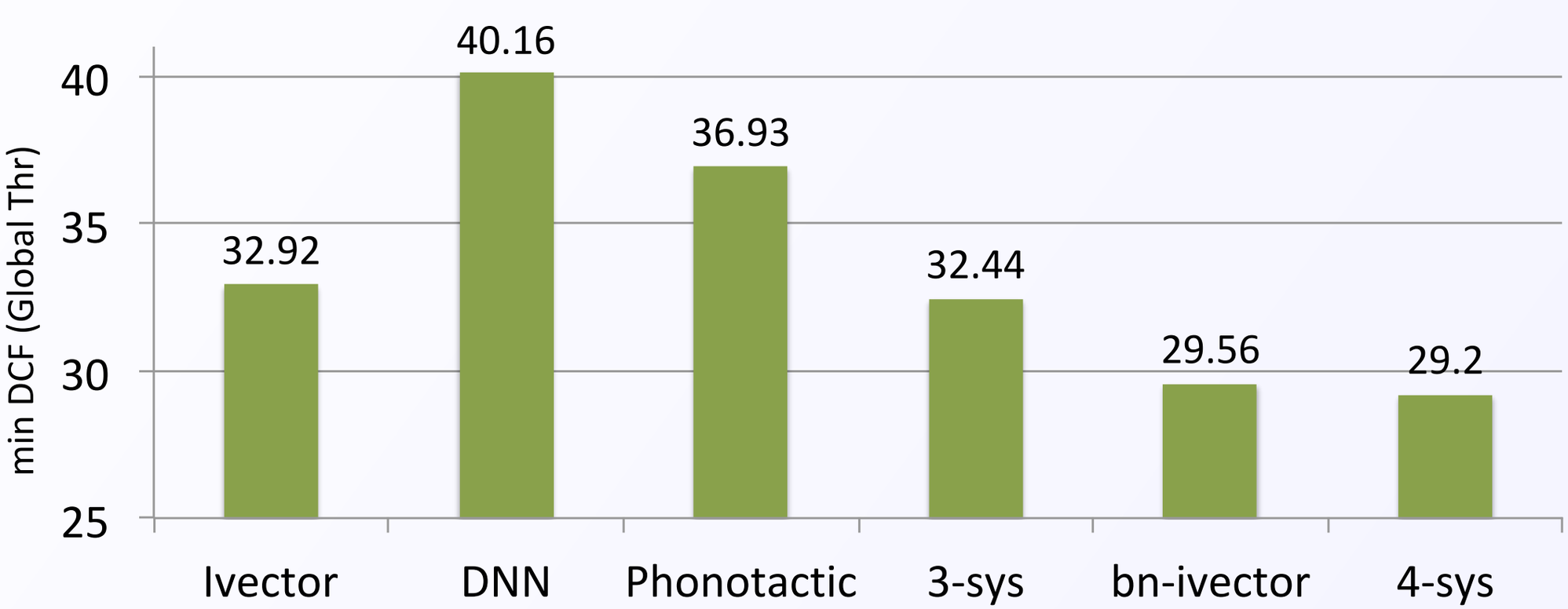
- Current state-of-the-art neural network approaches use NNs to extract embeddings, which are then scored by PLDA
- Using deep neural network in feature extraction



[Richardson, D. Reynolds, and Dehak 2015]

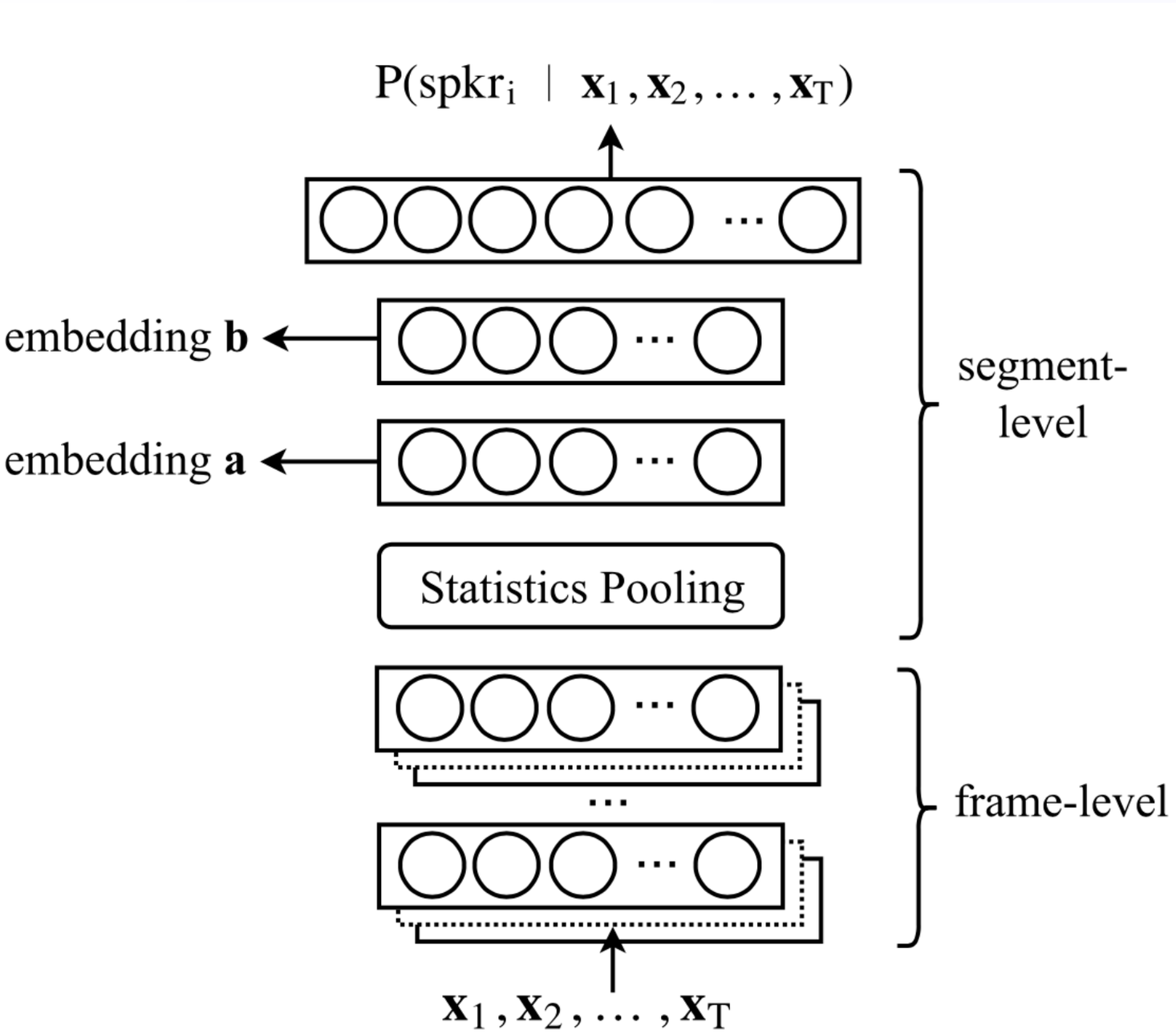
Our own work





[Ng, Nicolao, and Hain 2017]

X Vectors



Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	1500Tx3000
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512xN

Snyder et al 2018

References

Tomi Kinnunen and Haizhou Li. "An overview of text-independent spaeker recognition: From features to supervectors". In: *Speech Communication* 52 (2010), pp. 12–40.

Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. "Speaker Verification Using Adapted Gaussian Mixture Models". In: *Digital Signal Processing* 10.1-3 (Jan. 2000), pp. 19–41.

Michael E. Tipping and Christopher M. Bishop. "Mixtures of probabilistic principal component analysers". In: *Neural Computation* 11.2 (1999), pp. 443–482.

References

- JHL Hansen and T Hasan (2015), "Speaker Recognition by Machines and Humans: A tutorial review", IEEE Signal Processing Magazine, 32(6):74–99, https://ieeexplore.ieee.org/document/7298570
- E Variani et al (2014), "Deep neural networks for small footprint text-dependent speaker verification", ICASSP, https://ieeexplore.ieee.org/document/6854363
- D Snyder et al (2018), "X-Vectors: Robust DNN Embeddings for SpeakerRecognition", ICASSP, https://ieeexplore.ieee.org/document/8461375
- Raymond W. M. Ng, Mauro Nicolao, and Thomas Hain. "Unsupervised crosslingual adaptation of tokenisers for spoken language recognition". In: Computer, Speech and Language (2017).
- P. Kenny et al. "PLDA for speaker verification with utterances of arbitrary duration". In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. May 2013, pp. 7649–7653.
- F. Richardson, D. Reynolds, and N. Dehak. "Deep Neural Network Approaches to Speaker and Language Recognition". In: Signal Processing Letters, IEEE 22.10 (Oct. 2015), pp. 1671–1675.
- S. Furui. "Cepstral analysis technique for automatic speaker