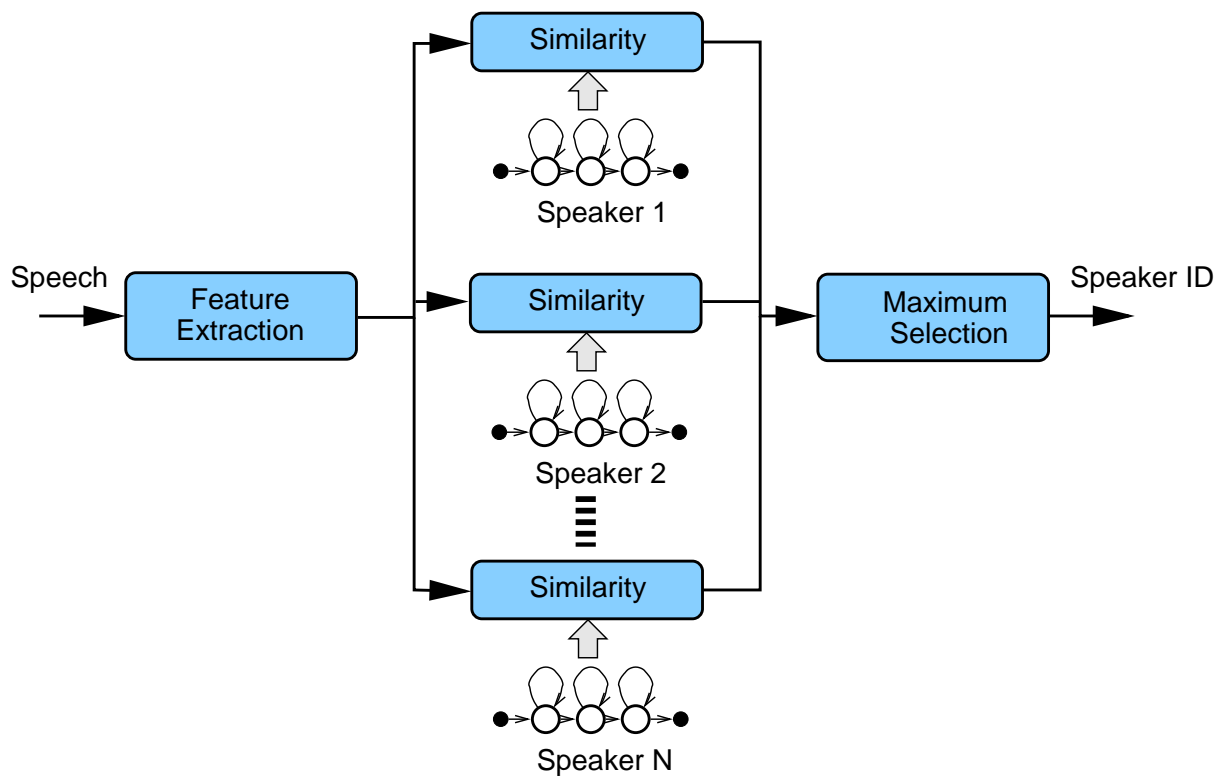


University of Cambridge
Engineering Part IIB

Paper 4F10: Statistical Pattern
Processing

Handout 12: Speaker Verification and
Identification



Mark Gales
mjfg@eng.cam.ac.uk
Michaelmas 2011

Introduction

This lecture looks at a particular application of statistical pattern processing, **speaker identification and verification**. These tasks can be summarised as

- **speaker identification**: *who am I?*
- **speaker verification**: *am I who I claim to be?*

The first is a multi-class problem (1 of K -classes), the second a binary classification problem (true/false).

Verification/identification is normally split into:

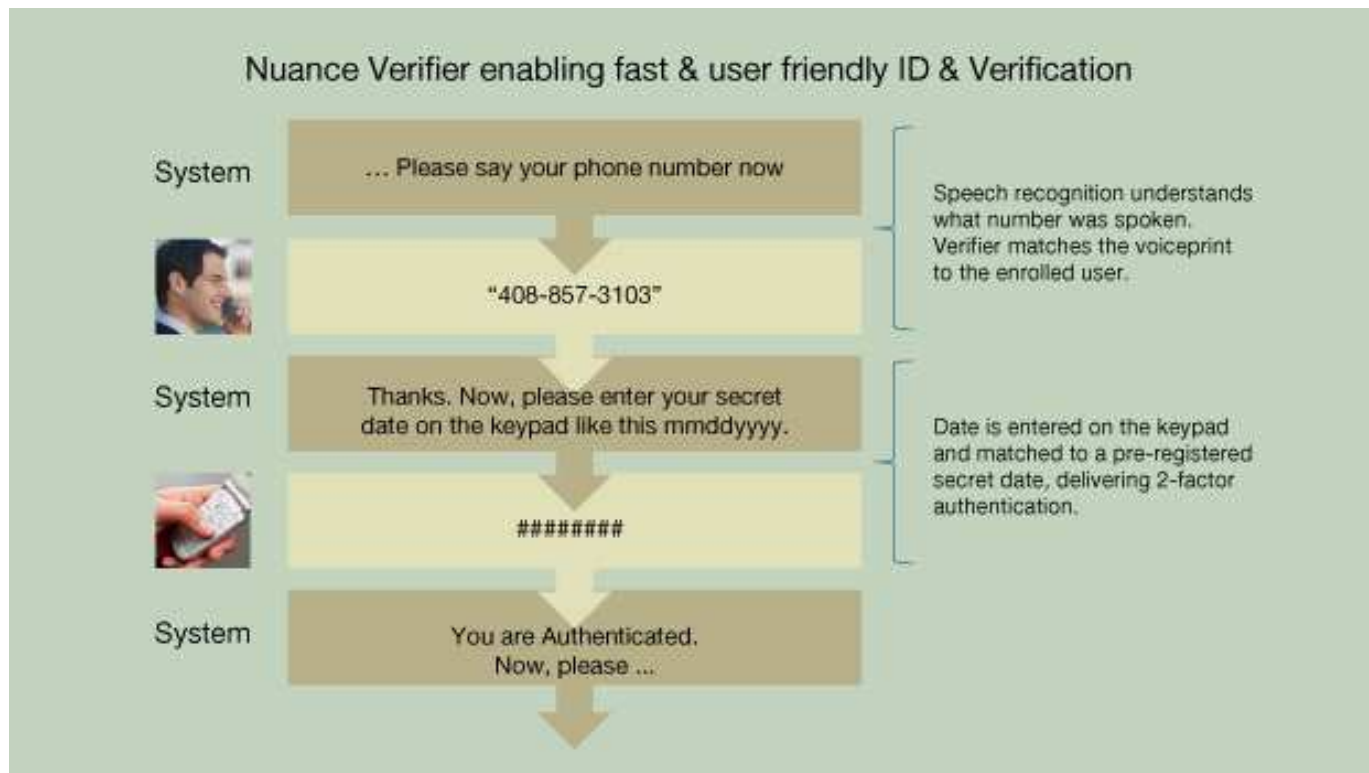
- **text dependent**: control over what the speaker will say;
- **text independent**: no control over what the speaker says
- **open/closed set** for identification

Example applications are:

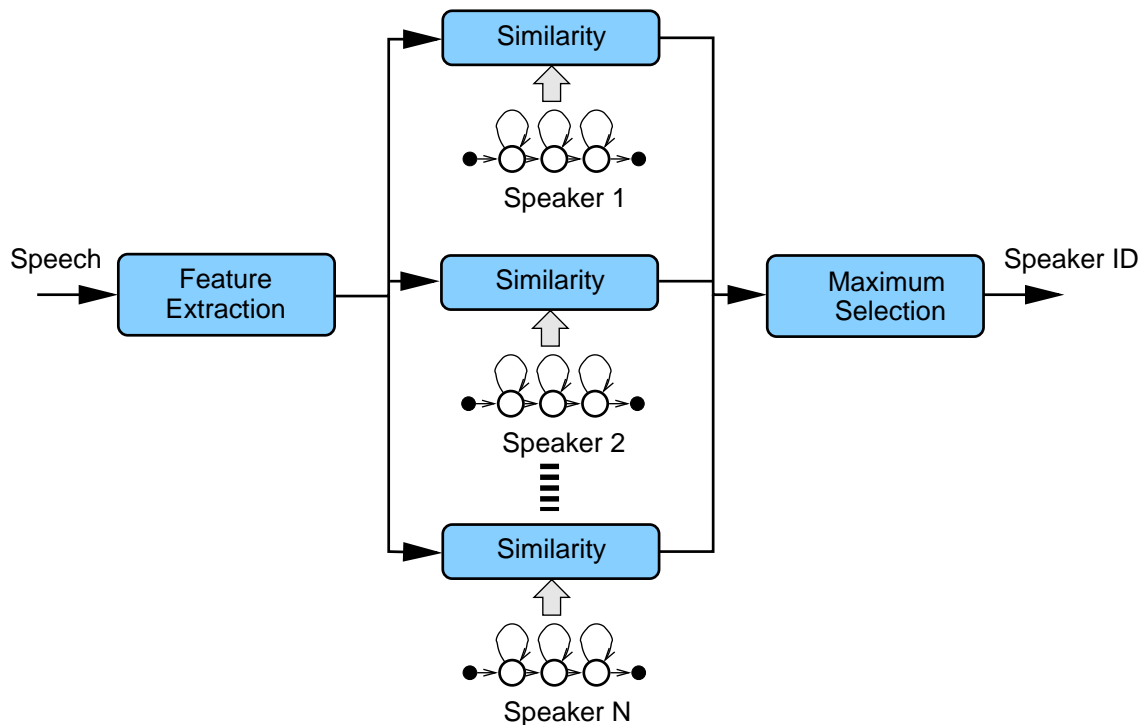
- **banking/shopping over the phone**: text-dependent, speaker verification;
- **forensic/security applications**: text-independent, speaker identification
- **speaker tracking in broadcast news transcription**: text independent, speaker identification.



Product - Nuance Verifier



Speaker Identification



A simple system for speaker identification is given above.

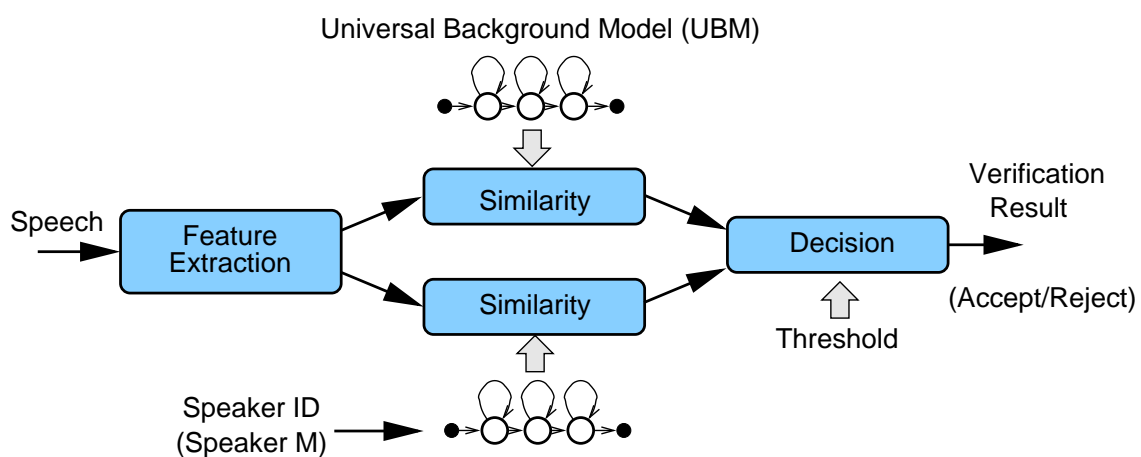
- **feature extraction:** reduce data rate from sampled signal (16KHz/16 bits)
 - 39 dimensional feature vector extracted each 10 ms
- **speaker model:** normally an Hidden Markov model/GMM
- **similarity measure:** likelihood measure
- **maximum selection:** Bayes' decision rule

This lecture will concentrate on text independent speaker verification.

Text Independent Speaker Verification

There are two stages of operation

- **Enrolment:** each speaker utters a small amount of speech (supervised training).
- **Verification:** a speaker claims an identity and utters a small amount of speech



The above diagram shows a standard speaker-verification system.

- **Universal Background Model:** a Gaussian mixture model
 - trained using all the speaker enrolment data
- **Speaker model:** one for each speaker
 - trained using the speaker-specific enrolment data

Bayes' decision rule is applied (equal priors)

$$P(\text{true}|\mathbf{O}, \boldsymbol{\theta}^{(s)}) = \frac{p(\mathbf{O}|\boldsymbol{\theta}^{(s)})}{\sum_{i=1}^S p(\mathbf{O}|\boldsymbol{\theta}^{(i)})} \approx \frac{p(\mathbf{O}|\boldsymbol{\theta}^{(s)})}{p(\mathbf{O}|\boldsymbol{\theta}^{\text{ubm}})}$$

Issues Addressed

This lecture will look at the following issues:

- **UBM model training**: an application of EM training
- **robust speaker model estimation**: an application of MAP estimation
- **performance assessment**: ROC/DET curves, Equal Error Rates (EERs)
- **SVMs for verification**: discriminative classifier
 - **dynamic kernels** to map from variable length data to a fixed length.

This lecture will **not** examine:

- details of feature extraction
- how changes in background environment are dealt with
- methods for increasing computational speed
- methods for compact speaker model representations
- HMMs for text-dependent modelling
- etc ...

Universal Background Model

The first stage is to train the UBM - this is meant to be a model of all the speakers.

Gaussian Mixture Models (GMMs) are used

$$p(\mathbf{O}_{1:T}|\boldsymbol{\theta}^{\text{ubm}}) = \prod_{t=1}^T p(\mathbf{o}_t|\boldsymbol{\theta}^{\text{ubm}}) = \prod_{t=1}^T \left(\sum_{m=1}^M c_m^{\text{ubm}} \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_m^{\text{ubm}}, \boldsymbol{\Sigma}_m^{\text{ubm}}) \right)$$

- the training data is obtained from the **enrolment** data from each speaker
- **expectation maximisation** (EM) can be used to train the model. For the “new” model mean

$$\hat{\boldsymbol{\mu}}_m^{\text{ubm}} = \frac{\sum_{s=1}^S \sum_{t=1}^{T^{(s)}} P(m|\mathbf{o}_t^{(s)}, \boldsymbol{\theta}^{\text{ubm}}) \mathbf{o}_t^{(s)}}{\sum_{s=1}^S \sum_{t=1}^{T^{(s)}} P(m|\mathbf{o}_t^{(s)}, \boldsymbol{\theta}^{\text{ubm}})}$$

where $P(m|\mathbf{o}_t, \boldsymbol{\theta}^{\text{ubm}})$ determined using “old” model parameters, $\boldsymbol{\theta}^{\text{ubm}}$, and s indicates the speaker.

- **diagonal covariance matrices** often used
 - faster likelihood calculation
 - fewer model parameters ($d = 39$)
- M normally in the range 256-2024

Speaker Enrolment

To do verification a speaker-specific model is required

- normally about 30 seconds of enrolment data per speaker
 - 3000 frames of data (10 ms frame-rate)
 - assume 1024 Gaussian components to estimate
- many components will not be seen/rarely seen

Maximum A-Posteriori MAP estimation

- use a **prior** on the model parameters and maximise

$$\hat{\theta} = \arg \max_{\theta} \{ \log(p(\mathbf{O}_{1:T}|\theta)) + \log(P(\theta)) \}$$

Form and parameters of the prior, $P(\theta)$, required

Would like to use a **conjugate prior**

- posterior has the same form as the prior distribution
- distribution must have **sufficient statistics of a fixed dimension**
- not possible for a mixture model

In practice a product of (for reference):

- **normal-Wishart** density for the component parameters (μ_m, Σ_m)
- **Dirichlet** density for the component priors (c_m)

this is a conjugate prior to the **complete data-set**. Only the mean update will be considered

MAP Estimate of the Mean

A normal-Wishart density has the form (reference)

$$p(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m | \tilde{\boldsymbol{\theta}}_m) \propto |\boldsymbol{\Sigma}_m|^{-(\alpha_m - d)/2} \times \exp\left(-\frac{\tau_m}{2}(\boldsymbol{\mu}_m - \tilde{\boldsymbol{\mu}}_m)' \boldsymbol{\Sigma}_m^{-1}(\boldsymbol{\mu}_m - \tilde{\boldsymbol{\mu}}_m) - \frac{1}{2} \text{tr}(\boldsymbol{\Sigma}_m^{-1} \tilde{\boldsymbol{\Sigma}}_m)\right)$$

where $\tilde{\boldsymbol{\theta}}_m$ are the parameters of the prior for component m

- $\alpha_m > d - 1, \tau_m > 0$
- $\tilde{\boldsymbol{\mu}}_m$ is a vector for component m (mean)
- $\tilde{\boldsymbol{\Sigma}}_m$ is a positive definite matrix for component m (covariance)

Only considering the mean updates (common in verification)

$$\hat{\boldsymbol{\mu}}_m^{(s)} = \frac{\tau_m \tilde{\boldsymbol{\mu}}_m + \sum_{t=1}^{T^{(s)}} P(m | \mathbf{o}_t^{(s)}, \boldsymbol{\theta}) \mathbf{o}_t^{(s)}}{\tau_m + \sum_{t=1}^{T^{(s)}} P(m | \mathbf{o}_t^{(s)}, \boldsymbol{\theta})}$$

This is an iterative process where $P(m | \mathbf{o}_t^{(s)}, \boldsymbol{\theta})$ is determined using the old model parameters, $\boldsymbol{\theta}$.

Where to get the prior parameters?

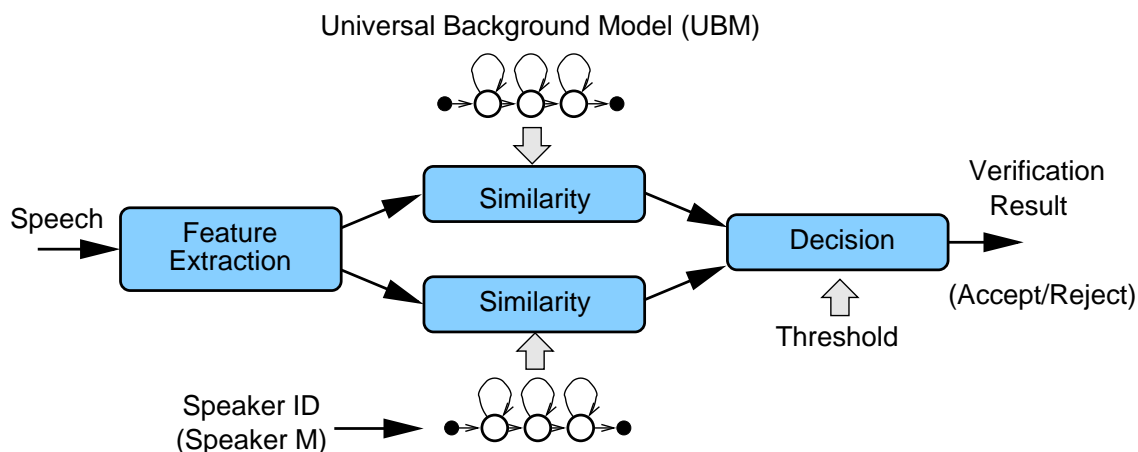
- prior mean is taken from the UBM - $\tilde{\boldsymbol{\mu}}_m = \boldsymbol{\mu}_m^{\text{ubm}}$
- τ_m is fixed for all components (normally set to 10-50)

Standard MAP attributes

- as $T^{(s)} \rightarrow \infty$ tend to ML estimate
- as $T^{(s)} \rightarrow 0$ tend to prior estimate

Speaker Verification

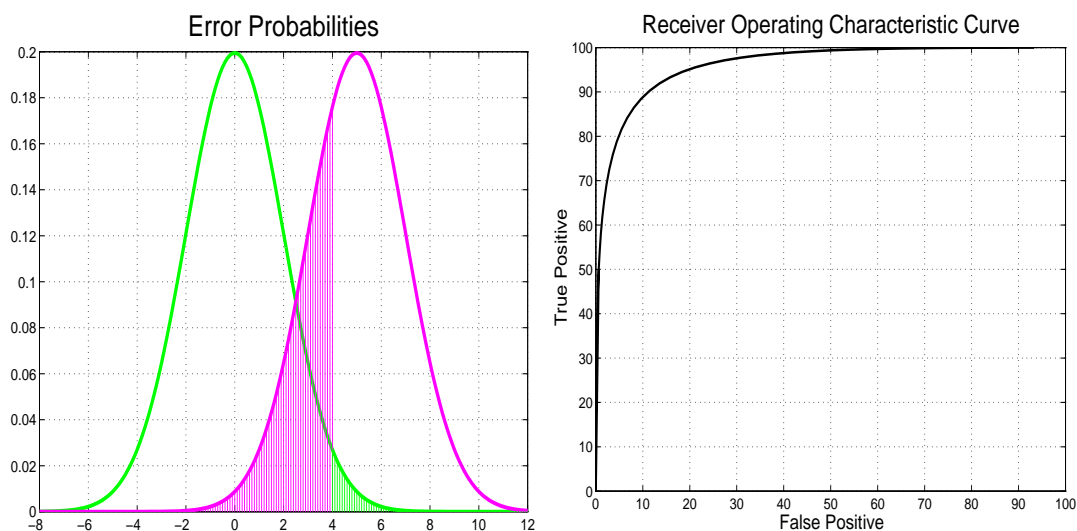
From the enrolment stage we have all the models



For verification use decisions of the form

$$\log(p(\mathbf{O}|\theta^{(s)})) - \log(p(\mathbf{O}|\theta^{\text{ubm}})) \begin{matrix} \text{true} \\ > \\ < \\ \text{false} \end{matrix} b$$

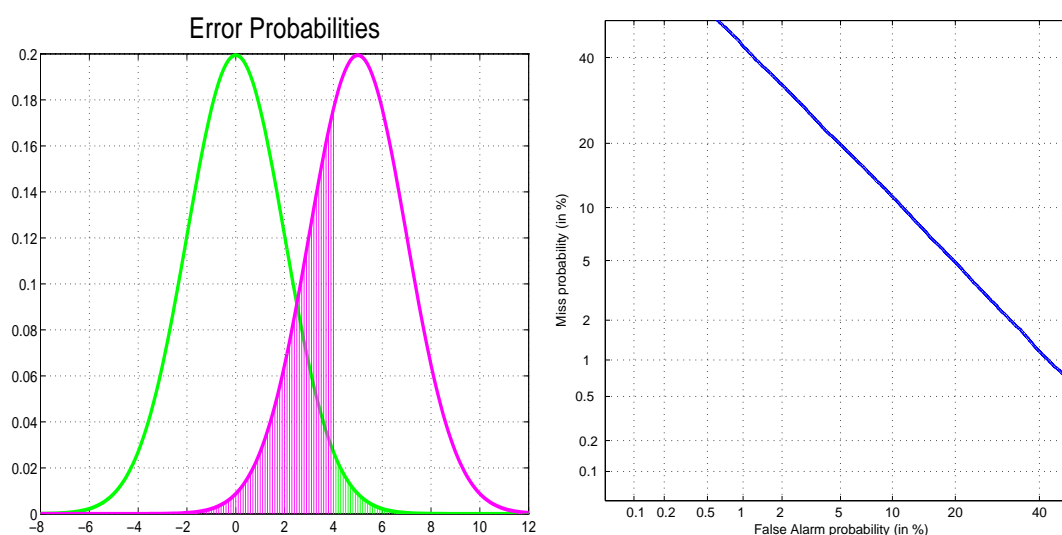
The setting of b may be determined using ROC curves



- b is set high for banking applications!

DET Curves and EER

Detection Error Trade-off (DET) curve is often used instead of a ROC curve for verification (and other **biometric** tasks).

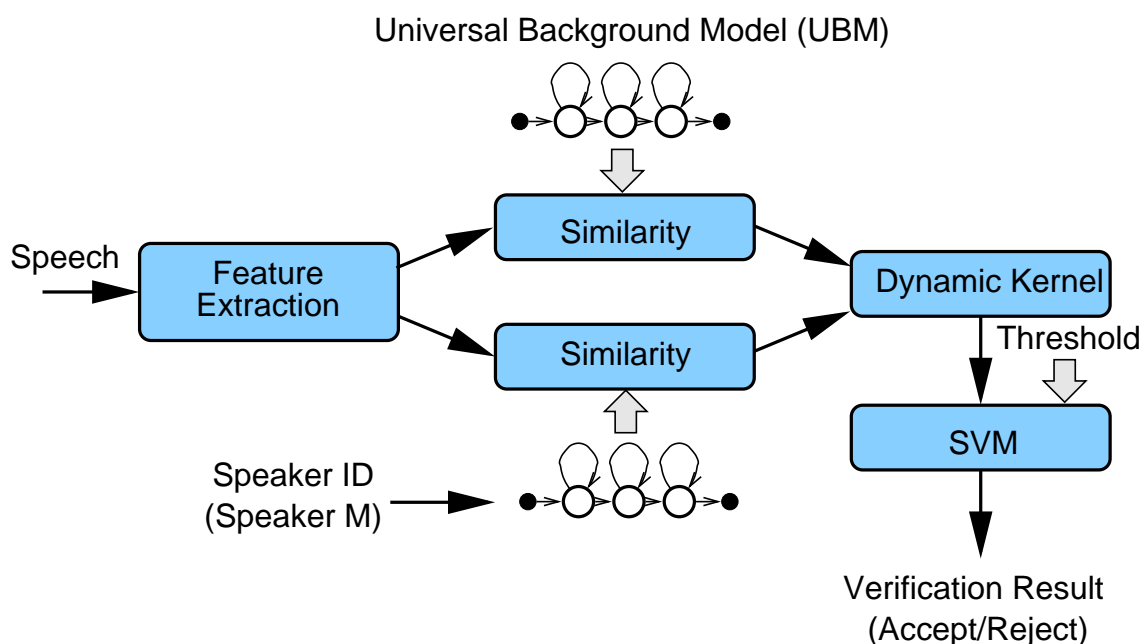


One problem with ROC is that the desired operating points are close together in the top left-hand corner. To modify this plot **miss probability to false-alarm probability** and use a mapped axis measured in standard deviates. This converts the previous plot into a straight line. Same information, but more clearly presented.

It is also useful to have a single number associated with system performance. **Equal error rate** is sometimes quoted: false-alarms equals false accepts. EER from sketch about 11%.

SVM-Based Verification

GMM-based speaker verification works well, but there has been interest in applying discriminative approaches to verification. One popular form is to use a **Support Vector Machine**

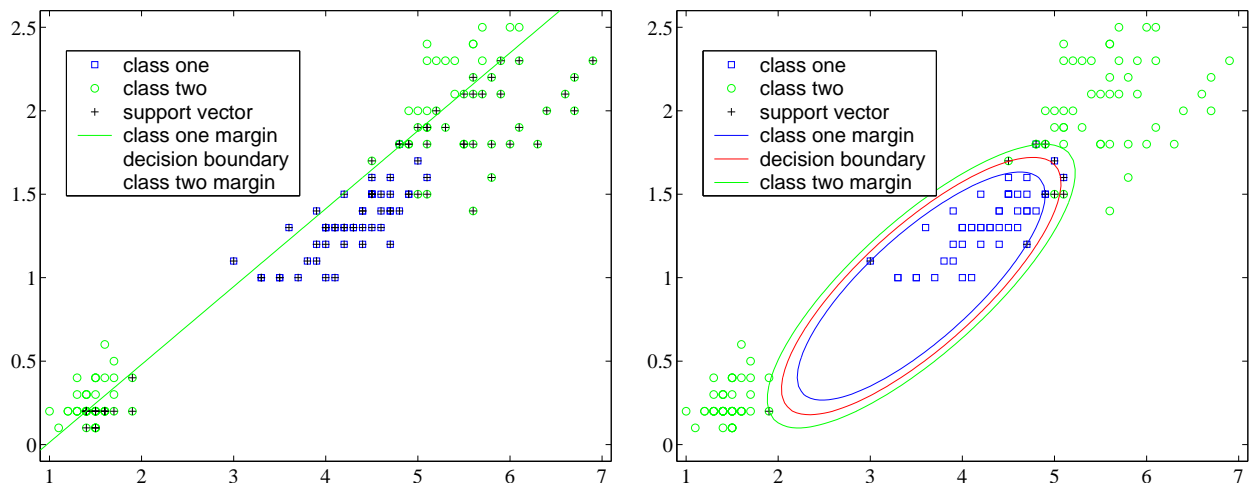


Verification (and other speech processing applications) awkward for direct application of SVMs

- the “observation” ($O_{1:T}$) size will vary from speaker to speaker
- cannot directly use observations in SVM
- **dynamic kernels** are one approach to handling this

Dynamic Kernels

Kernels often used with SVMs



- SVM decision boundary linear in the feature-space
 - make non-linear using a non-linear mapping $\phi()$ e.g.

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}, \quad k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

- Efficiently implemented using a **Kernel**: $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^2$

Applying SVMs to speech data awkward

- speech data varies in length
- could sub-sample data, but loses information
- **Dynamic Kernels** offer a solution
 - map variable length data to a fixed length
 - standard SVM training can then be used

Handling Sequence Data

Sequence data has **inherent variability** in number of samples:

- speech data at a fixed frame rate
- DNA/protein sequences

The	cat	sat	on	the	mat
-----	-----	-----	----	-----	-----

 1200 frames

$$\mathbf{O}_{1:1200}^{(1)} = \{\mathbf{o}_1, \dots, \mathbf{o}_{1200}\}$$

The	cat	sat	on	the	mat
-----	-----	-----	----	-----	-----

 900 frames

$$\mathbf{O}_{1:900}^{(1)} = \{\mathbf{o}_1, \dots, \mathbf{o}_{900}\}$$

Dynamic kernels map these sequences to a fixed length

- allows standard SVM training to be used
- hopefully make use of all the data

The simplest feature-space is to use the log-likelihood

$$\phi(\mathbf{O}_{1:T}) = [\log(p(\mathbf{O}_{1:T}|\boldsymbol{\theta}))]$$

How to increase the dimensionality sensibly?

Fisher Kernels

Fisher kernels are one example of a dynamic kernel. They make use of a **generative model**, $p(\mathbf{O}_{1:T}|\boldsymbol{\theta})$ and are defined as

$$k(\mathbf{O}_{1:T(i)}^{(i)}, \mathbf{O}_{1:T(j)}^{(j)}) = \boldsymbol{\phi} \left(\mathbf{O}_{1:T(j)}^{(j)} \right)' \boldsymbol{\Sigma}^{-1} \boldsymbol{\phi} \left(\mathbf{O}_{1:T(j)}^{(j)} \right)$$

where

$$\boldsymbol{\phi}(\mathbf{O}_{1:T}) = \nabla_{\boldsymbol{\theta}} \log (p(\mathbf{O}_{1:T}|\boldsymbol{\theta}))|_{\hat{\boldsymbol{\theta}}}$$

and $\boldsymbol{\Sigma}^{-1}$ is a positive definite matrix that defines the **metric** for the feature-space (this has been taken as an identity matrix so far). More generally it is defined as (assuming zero mean)

$$\boldsymbol{\Sigma} = \mathcal{E} \{ \boldsymbol{\phi}(\mathbf{O}) \boldsymbol{\phi}(\mathbf{O})' \} = \int \boldsymbol{\phi}(\mathbf{O}) \boldsymbol{\phi}(\mathbf{O})' p(\mathbf{O}|\boldsymbol{\theta}) d\mathbf{O}$$

This is the **Fisher Information Matrix**

Considering just the means of a GMM

$$\boldsymbol{\phi}(\mathbf{O}_{1:T}) = \begin{bmatrix} \sum_{t=1}^T P(1|\mathbf{o}_t, \hat{\boldsymbol{\theta}}) \hat{\boldsymbol{\Sigma}}_1^{-1} (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_1) \\ \vdots \\ \sum_{t=1}^T P(M|\mathbf{o}_t, \hat{\boldsymbol{\theta}}) \hat{\boldsymbol{\Sigma}}_M^{-1} (\mathbf{o}_t - \hat{\boldsymbol{\mu}}_M) \end{bmatrix}$$

This is a $M \times d$ features vector.

This kernel can be trained on large amounts of unlabelled data, the classifier is then trained on a small amount of labelled training data.

Generative Kernels

A modified version of the Fisher Kernel, the **generative kernel**, can be used for speaker verification.

One form is

$$\phi(\mathbf{O}_{1:T}) = \begin{bmatrix} \log(p(\mathbf{O}_{1:T}|\boldsymbol{\theta}^{(s)})) - \log(p(\mathbf{O}_{1:T}|\boldsymbol{\theta}^{\text{ubm}})) \\ \nabla_{\boldsymbol{\theta}} \log(p(\mathbf{O}_{1:T}|\boldsymbol{\theta}))|_{\boldsymbol{\theta}^{(s)}} \end{bmatrix}$$

- 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

An SVM is trained for **each** speaker (interestingly only one positive training example works!). Verification is then based on

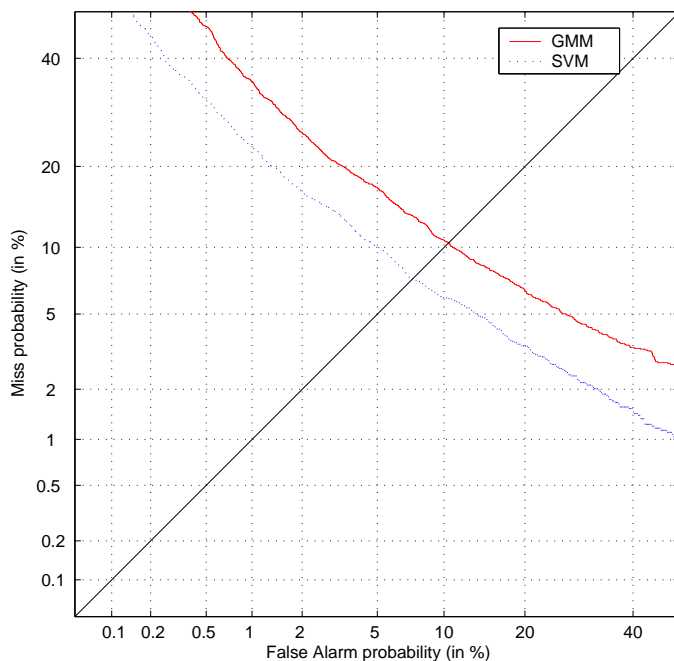
$$\langle \mathbf{w}, \phi(\mathbf{O}_{1:T}) \rangle \begin{array}{l} \text{true} \\ > \\ < \\ \text{false} \end{array} b$$

where \mathbf{w} is the decision boundary obtained from training the SVM.

Example Task

As an example of speaker verification the NIST 2002 Speaker Recognition evaluation

- utterances recorded over a cellular network
- 139 male, 191 female speakers
- 1 enrolment utterance/speaker (upto 2 mins)
- 3570 test utterances (90% imposters)



MAP (τ)	EER	
	GMM	SVM
0	11.94	9.54
10	10.43	7.75
25	11.37	7.31
50	12.33	7.44

SVM-based verification outperforms GMM-based systems.