

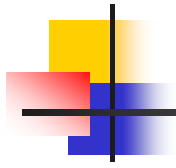


Speaker Recognition

Luciana Ferrer

SRI International

Stanford University



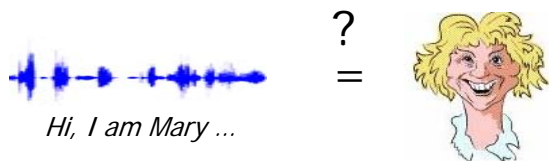
Agenda

- Introduction
- Baseline GMM-UBM system
- Improving the baseline
 - Other cepstral systems
 - Stylistic systems
- Combination issues
- Conclusions

Speaker Recognition Tasks

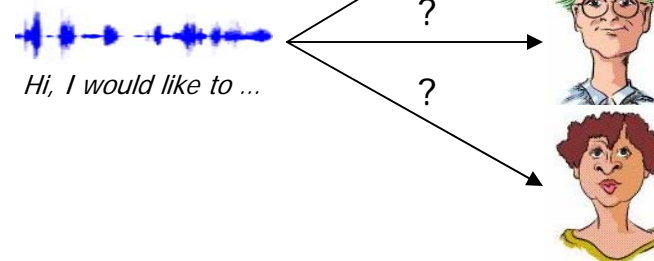
Speaker Verification

Is this Mary's voice?



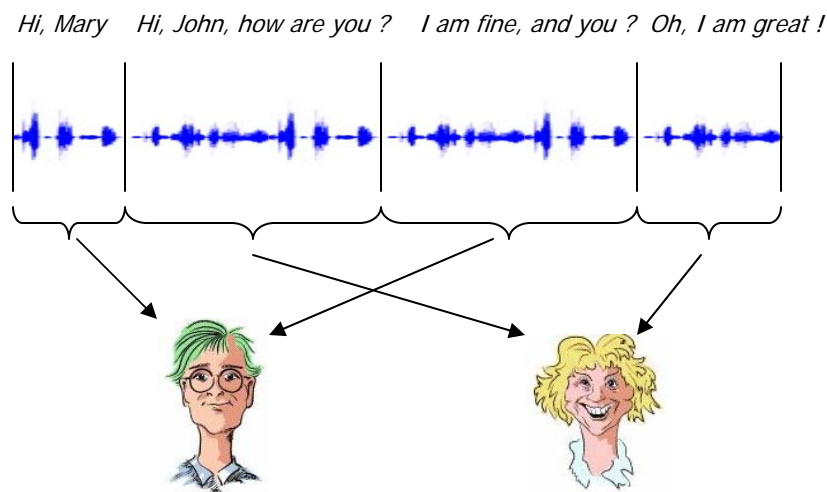
Speaker Identification

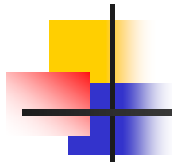
Whose voice is this?



Segmentation and clustering

Where speaker changes occur?
Who is speaking in each segment ?

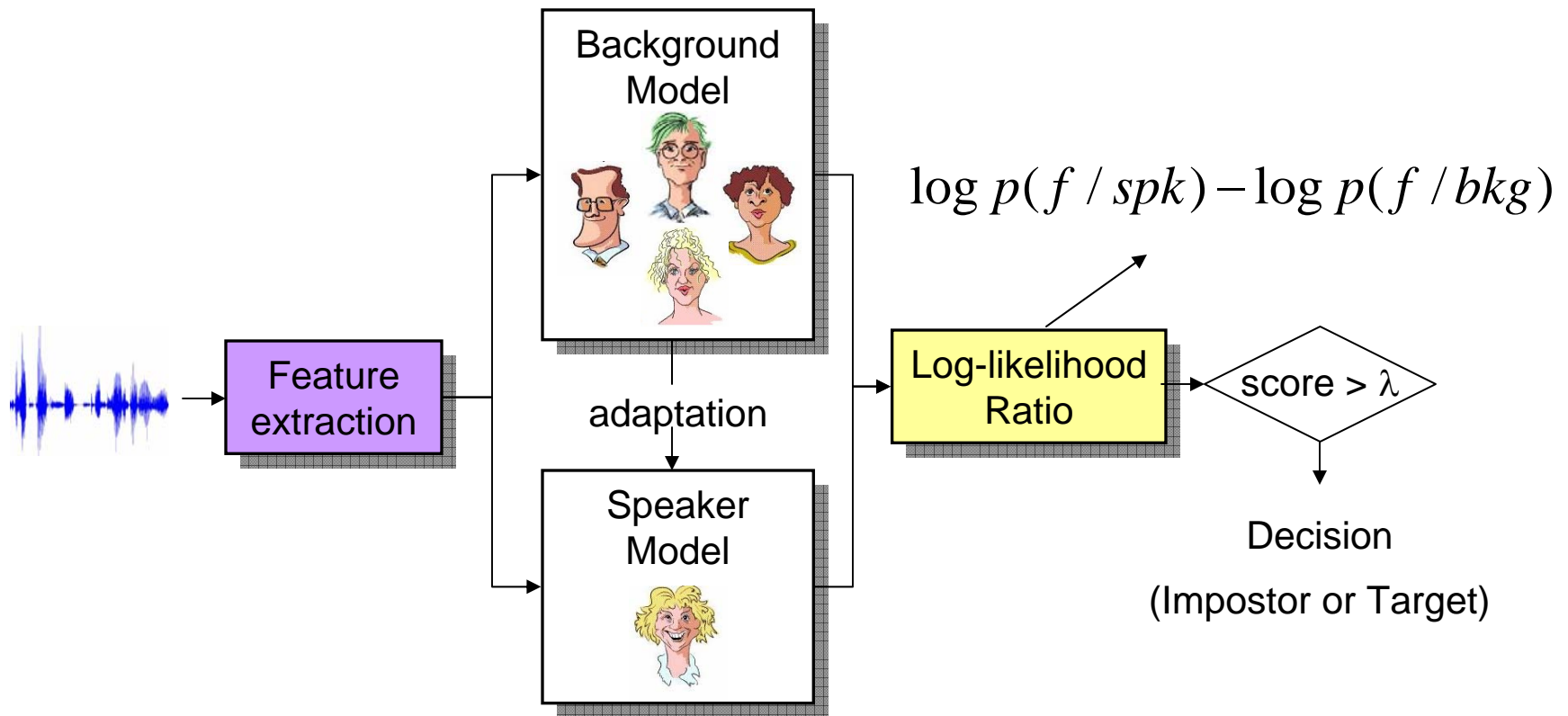




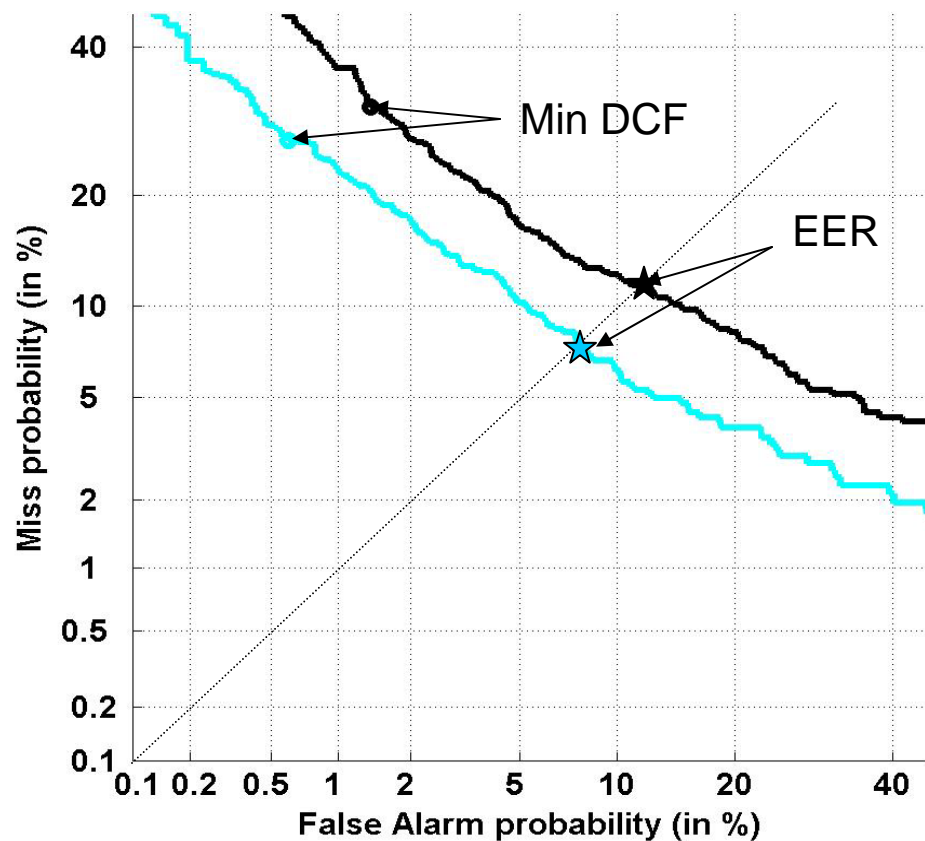
General Issues

- Text-dependent Vs text-independent
 - Text dependency simplifies the classification problem but constraints the usage of the system
- Amount of train and test data
 - Large amounts of train/test data allow for use of sparse features
- Variation between test and train conditions
 - Different channels or noise condition
 - Change in interlocutors, mood, health ...

Paradigm for speaker verification



Performance Measures



Equal Error Rate

$$EER = P_{\text{miss}} \text{ when } P_{\text{fa}} = P_{\text{miss}}$$

Detection Cost Function

$$C = \begin{cases} C_{\text{miss}} & \text{if miss} \\ C_{\text{fa}} & \text{if false alarm} \\ 0 & \text{otherwise} \end{cases}$$

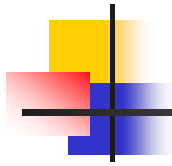
$$DCF = E[C] = E[E[C/\text{true class, det class}]]$$

$$DCF = C_{\text{miss}} P_{\text{target}} P_{\text{miss}} + C_{\text{fa}} P_{\text{imp}} P_{\text{fa}}$$

$$(C_{\text{miss}}=10, C_{\text{fa}}=1, P_{\text{target}}=0.01)$$

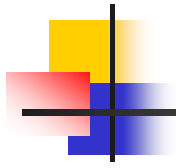
Misses: Targets classified as impostors

False alarms: Impostors classified as targets



Benchmark evaluations

- Yearly evaluations organized by NIST (National Institute of Standards and Technology)
- Task:
 - Speaker verification
 - Several conditions for different amounts of train/test data
- Databases:
 - Telephonic speech
 - Handsets used vary from conversation to conversation
- We are evaluated based on (actual) DCF
- Results in this talk:
 - 1- and 8-conv side training, 1-conv side testing condition
 - Each conv side has approx. 2.5 minutes of speech
 - more than 20,000 trials in each condition

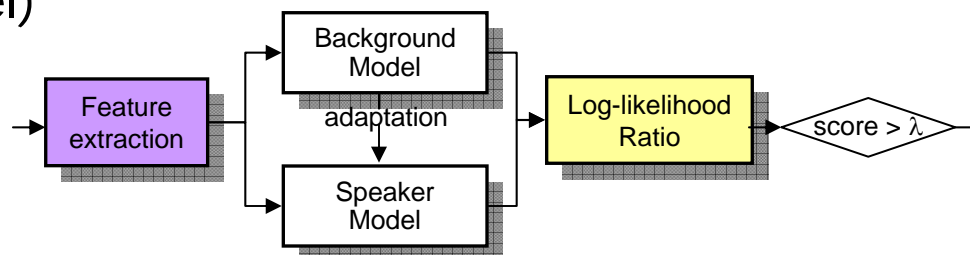


Agenda

- Introduction
- Baseline GMM-UBM system
- Improving the baseline
 - Other cepstral systems
 - Stylistic systems
- Combination issues
- Conclusions

Baseline System

- UBM-GMM paradigm: (Universal Background Model – Gaussian Mixture Model)

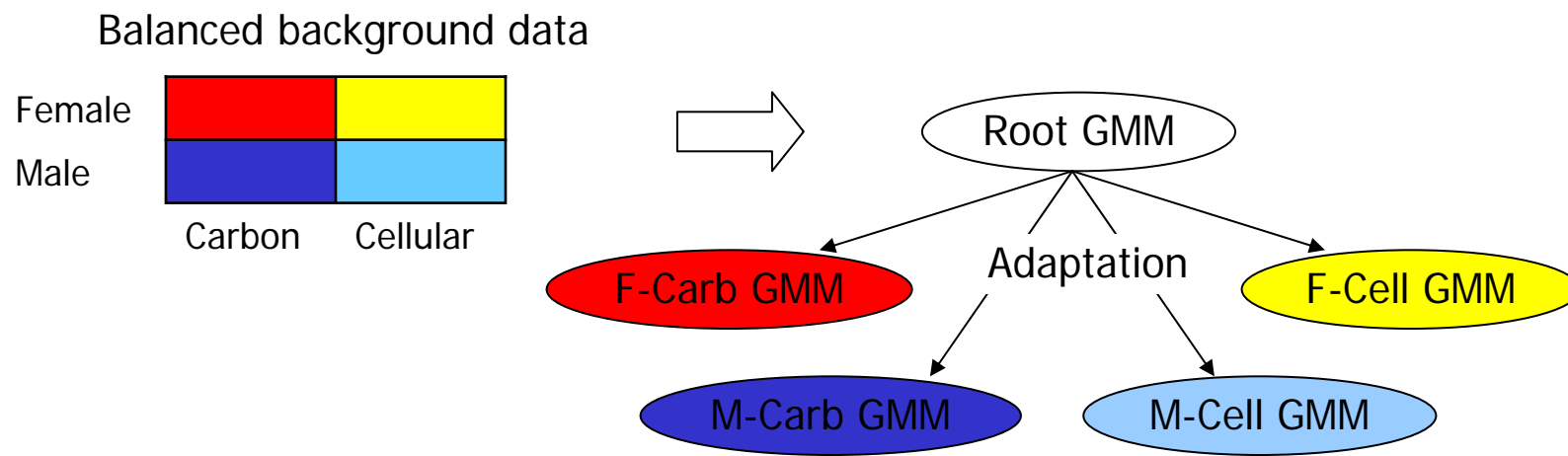


- Features:
 - Mel Frequency Cepstral Coefficients (MFCCs) with deltas, double deltas and triple deltas
- Training:
 - Background Model: train a GMM using thousands of speakers
 - Speaker Model: adapt the parameters of the GMM to the speaker's data
- Testing:
 - Compute likelihood ratio between both models
 - Threshold the obtained score

Baseline System - Improvements

Feature transformation (Reynolds 2003)

- Training:



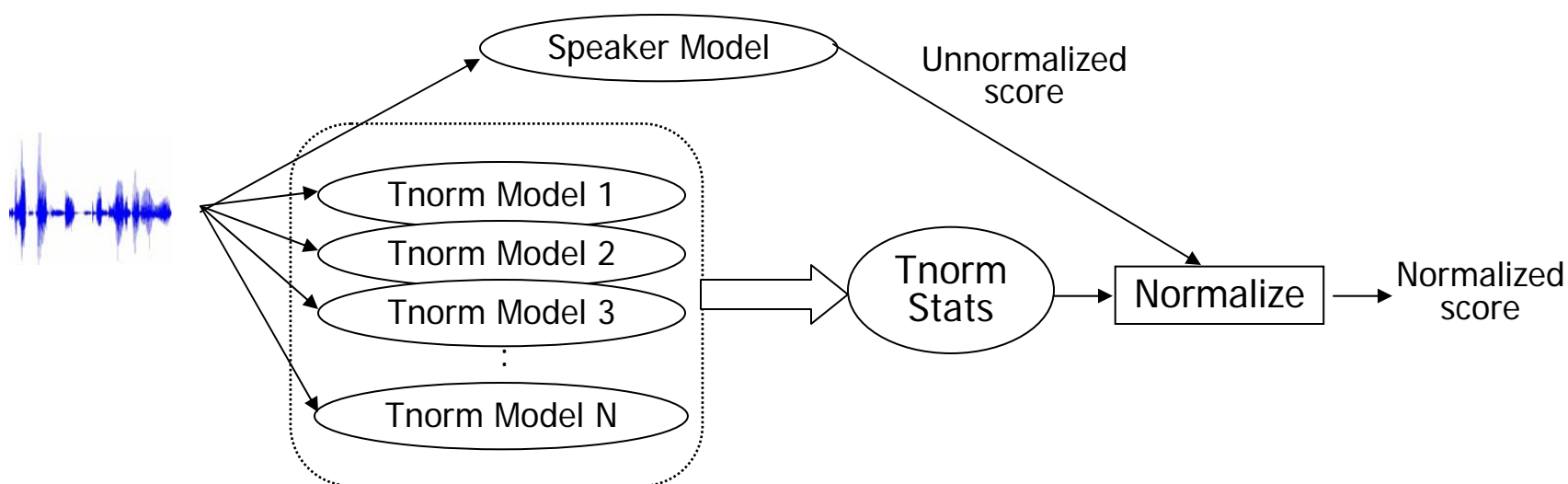
- Testing:

- Detect the most likely label for the test utterance
- Using corresponding model, transform each feature vector back to the root GMM
- End up (hopefully !) with channel and gender independent feature vectors

Baseline System - Improvements

Score Normalization

- TNorm (Test-length normalization): (Reynolds 1997)
 - Compensates for bias due to test length (and more)
 - Training: Create a balanced set of models for “impostor” speakers
 - Testing: Normalize the speaker’s score with the stats from the tnorm scores.

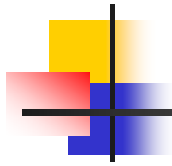




Baseline System - Improvements

Score Normalization

- Hnorm (Handset Normalization): (Reynolds 1997)
 - Compensates for bias due to handset mismatch.
 - Training: test the target model against data from each handset and gender (impostors). Generate stats for each label.
 - Testing: Detect label for test utterance and normalize score accordingly.
 - Replaced by feature normalization.
- Znorm:
 - Same as Hnorm but without splitting the impostor scores in sets.
 - Does not add much on top of feature normalization and Tnorm.



Baseline System – Our results

Results by training condition

System	1-conv		8-conv	
	% EER	DCF	% EER	DCF
Baseline (No Tnorm)	7.22	0.297	5.33	0.204
Baseline	7.17	0.248	4.91	0.169

- With feature transformation and Tnorm.
- 2048-component GMM trained with hundreds of hours of speech
- **Warning:** Results are highly dependent on database and task



Agenda

- Introduction
- Baseline GMM-UBM system
- Improving the baseline
 - Other cepstral systems
 - Stylistic systems
- Combination issues
- Conclusions



Improving the baseline

- Baseline relies on short-time acoustic information only
 - Suffers under noisy conditions or handset mismatch
 - No order information
 - Stylistic information missing:
 - Habitual word patterns (you know, like, as a matter of fact)
 - Prosodic patterns (e.g., pausing, timing, intonation)
 - Discourse-related patterns (e.g., turn-taking)
- First attempts to include longer range information started around 1997
- Success of these new systems is measured based on how much they improve performance in combination with baseline
 - 2002 JHU Workshop: huge improvements over baseline (from 0.7% EER to 0.2% on 8-conv training condition)



MLLR Cepstral System

- Motivation: normalize out text dependency in cepstral speaker modeling
- MLLR (Maximum Likelihood Linear Regression) Transforms:
 - Affine mapping of Gaussian means during speaker adaptation in ASR
 - Turns *speaker-independent* into *-dependent* models
 - Uses phone-loop model or prior recognition output
- Idea: use MLLR coefficients as feature vectors and model with SVMs
- Side benefit: ASR front-end and feature transforms normalize out channel effects



MLLR Cepstral System

- Features: Combine MLLR transforms from two ASR stages:
 - 1st stage: MFCC, 2 phone classes, adapt to phoneloo model
 - 2nd stage: PLP, 8 phone classes, adapt to 1st recognition hyps
 - Discard nonspeech transforms
- Rank Normalization:
 - Some models are sensitive to relative scaling of feature dimensions
 - Absent prior knowledge, ranges should be roughly equal on all dimensions
 - Rank-norm: replace each sample by its rank in background distribution

Background data:	.34	.35	4.3	5.6	100
Data point:				7	
Rank-normalized:	0.2	0.4	0.6	0.8	1.0

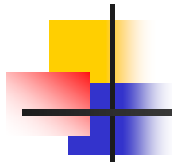
- Maps reference distribution to a uniform distribution [0 ... 1]
- Distance between two feature values = percentage of population that lies between them



MLLR Cepstral System - Results

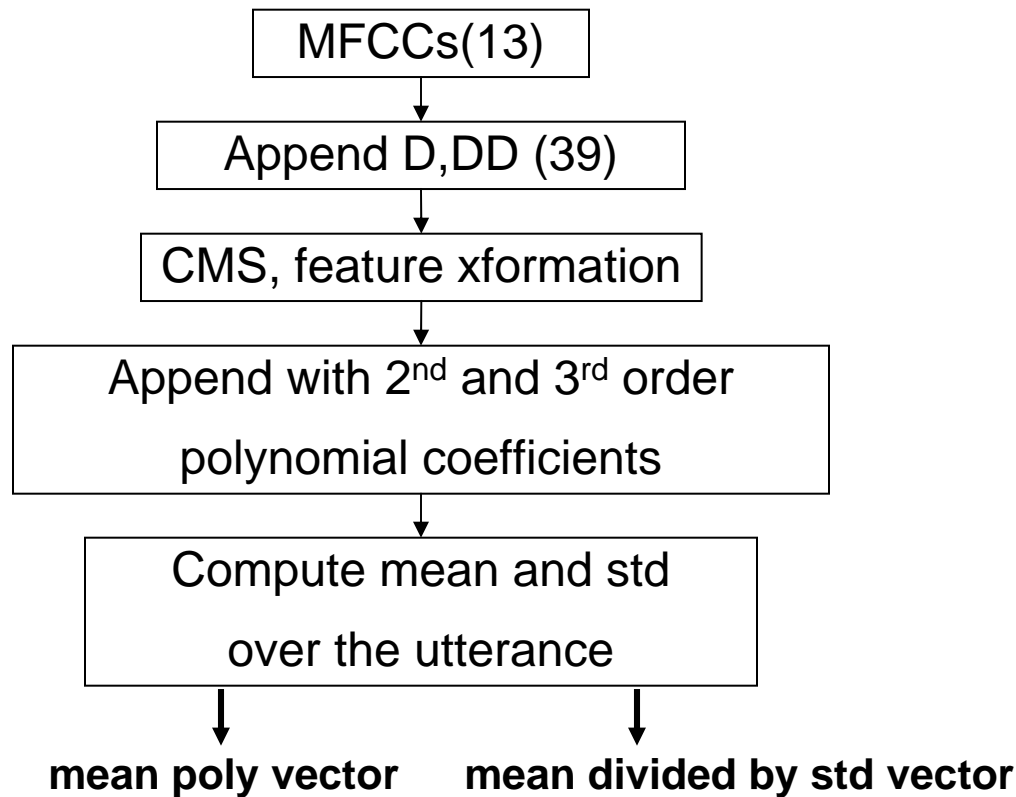
System	1-conv		8-conv	
	% EER	DCF	% EER	DCF
Baseline	7.17	0.248	4.91	0.169
MLLR	10.33	0.252	5.50	0.120
Baseline + MLLR	4.93	0.174	3.23	0.096

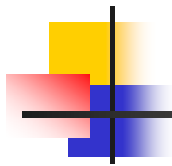
- Even though it uses the same initial features as the baseline, it helps in combination
- It performs really well on 8 conv training. Baseline cannot make good use of so much training data.



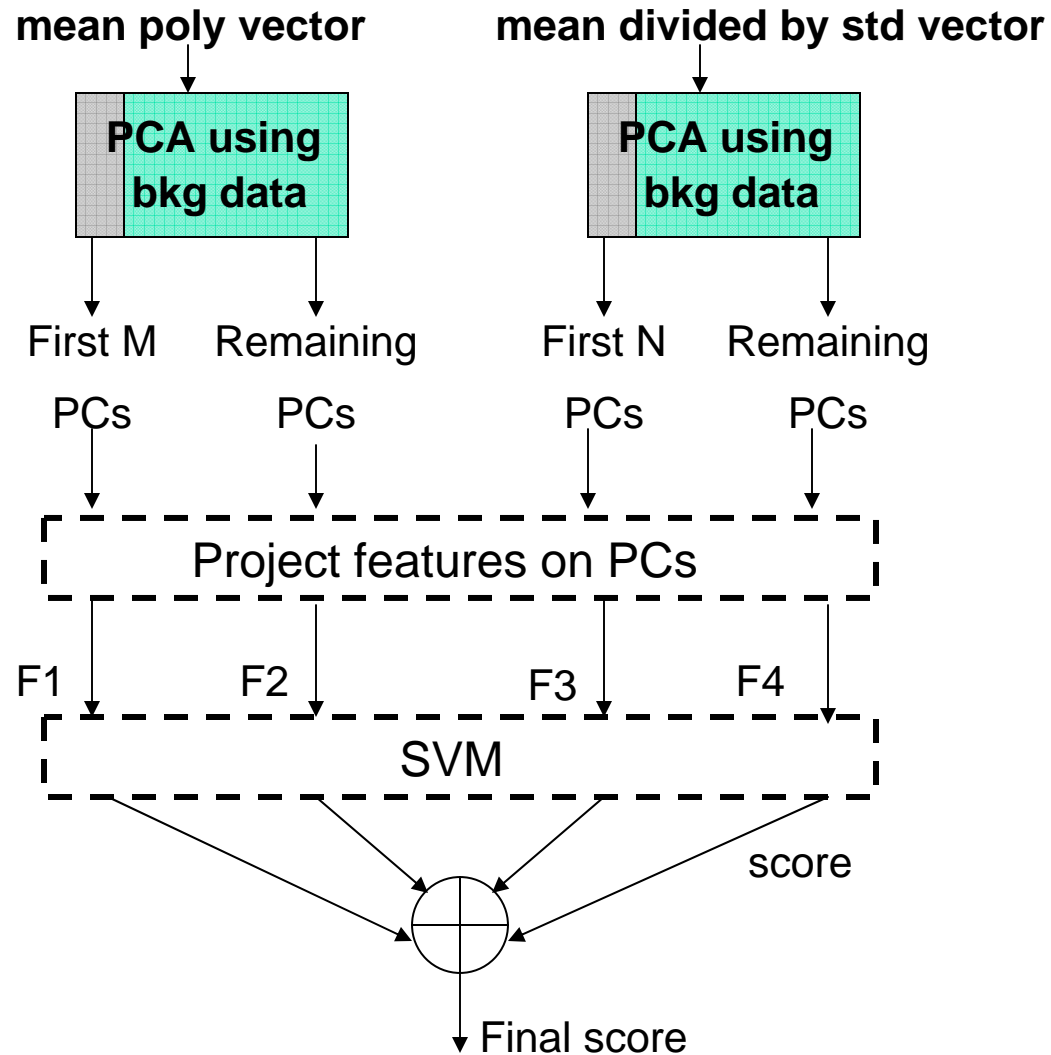
MFCC-SVM System

- MFCC features with lots of processing:





MFCC-SVM System





MFCC-SVM System - Results

System	1-conv		8-conv	
	% EER	DCF	% EER	DCF
Baseline	7.17	0.248	4.91	0.169
MLLR	10.33	0.252	5.50	0.120
MFCC-SVM	7.26	0.269	3.05	0.103
Baseline + MLLR	4.93	0.174	3.23	0.096
Baseline + MFCC-SVM	5.82	0.218	2.99	0.100

- Even though it is better than the MLLR system individually, it is worse in combination with baseline
 - MLLR system is actually using extra information (alignments)



Conditional Phone Pronunciations

- Developed at JHU workshop
- Aim: Learn speaker-dependent pronunciations by matching constrained ASR phones with open-loop alignments from different languages
- Training: Align ASR word phones with open-loop (OL) phones at frame level and compute conditional probabilities

TIME	ASR	EG	GE	SP	JA	MA
24964	t	n	n	n	sh	N
24965	t	s	h	s	sh	N
24966	t	s	h	s	sh	N
24967	t	s	h	s	sh	S
24968	t	s	h	s	sh	S
24969	t	s	h	s	sh	S
24970	t	s	h	s	rx	S
24971	ax	l	h	s	rx	i:
24972	ax	l	h	iy	rx	i:
24973	ax	l	h	iy	y	i:

$$\Pr(OLph / ASRph, Spkr) = \frac{\#(OLph, ASRph)}{\# ASRph}$$

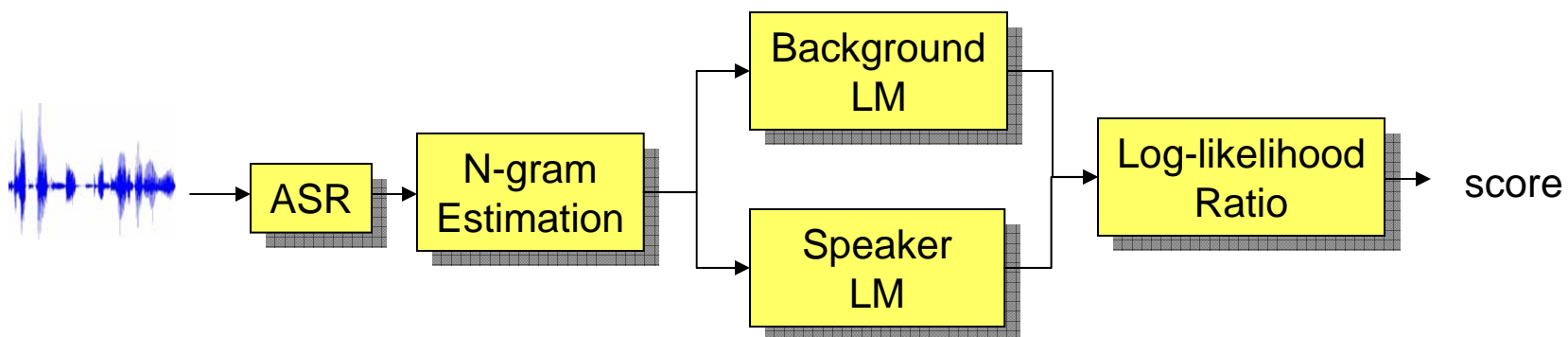


Conditional Phone Pronunciations

- Testing: Compute likelihood of observed (OL_phone,ASR_phone) sequence against speaker and background model
- Scores from five OL phone streams linearly combined
- Performed amazingly close to baseline, but did not combine so well
- Requires phone recognizer for several different languages

Word N-Gram Language Model

- Goal: Exploit idiosyncratic usage of words and word sequences (Doddington'01)
- Features: Frequencies of word bigrams in the conv-side
 - Example: "uh uh", "you know", "you bet", "sort of", "in terms of"
- Vocabulary fixed as the N most frequent bigrams in text
- Training: model is a collection of probabilities given by the bigram frequencies in train data
- Testing: score is log-likelihood ratio of features given background and speaker's models.





Word N-Gram SVM

- Features are, again, relative frequencies of word N-grams in the conv-side
 - But now take unigram, bigrams and trigrams
- Model this vector of features using an SVM
 - SVMs can deal with tons of features when the vectors are sparse
- Performs better than LM version



Word N-Gram SVM - Results

System	1-conv		8-conv	
	% EER	DCF	% EER	DCF
Baseline	7.17	0.248	4.91	0.169
MLLR	10.33	0.252	5.50	0.120
Word SVM	24.58	0.860	11.25	0.484
Baseline + MLLR	4.93	0.174	3.23	0.096
Baseline + Word SVM	6.42	0.228	3.23	0.123

- Individual system is much worse than cepstral systems, but in combination adds almost as much as them
- Performance of this system relative to baseline is better for 8-conv training



Prosodic Features

- Aim: Capture idiosyncrasies in the use of duration, energy and pitch patterns
- Tools
 - ASR: word, phone, state level alignments
 - get_f0: extraction of energy and pitch values per frame
 - Scripts, scripts and more scripts ...
- Features
 - Compute “stylized” energy and pitch profiles
 - Compute elaborated features from alignments, raw energy and pitch values and those profiles

AlgeMy: SRI's feature computation software

The screenshot displays the AlgeMy software interface, which is used for feature computation. The window title is "AlgeMy: C:\Documents and Settings\lferrer.SPEECH\My Documents\eclipse workspace\algeMy\snerfs.alg". The interface includes a menu bar (File, Edit, View, Run, Help), a toolbar with icons for file operations and execution, and a main workspace divided into several panels.

Table Panel: A table with two columns: "ID" and "#Frames". The data is as follows:

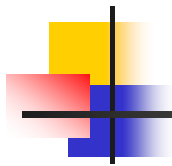
ID	#Frames
sw24292B	29996
sw24293A	29996
sw24293B	29996
sw24294A	29996
sw24294B	29996
sw24295A	29996
sw24295B	29996
sw24296A	29996
sw24296B	29996

Plot Panel: A line graph showing a signal over time. The x-axis is labeled from 2.150 to 2.200 with a multiplier of 10^4 . The y-axis is labeled from 1 to 5 with a multiplier of 10^2 . The plot shows a fluctuating signal with several peaks. Controls for "Zoom out to fit" and "Undim" are visible.

Flowchart Panel: A complex flowchart titled "Definition of Regions" and "Pitch and Energy Features". It consists of numerous interconnected nodes representing different processing steps, such as "Align Reader", "Pass Filter", "Merge Pauses", "String Concat", "String Paste", "String PS Filter", "Spell View Spell", "Spell NonView Spell", "Pitch Confidence Filter", "Vowels RS", "Median Filter", "Filter LV", "Stylizer", "Line -> Vector", and "Region Merge".

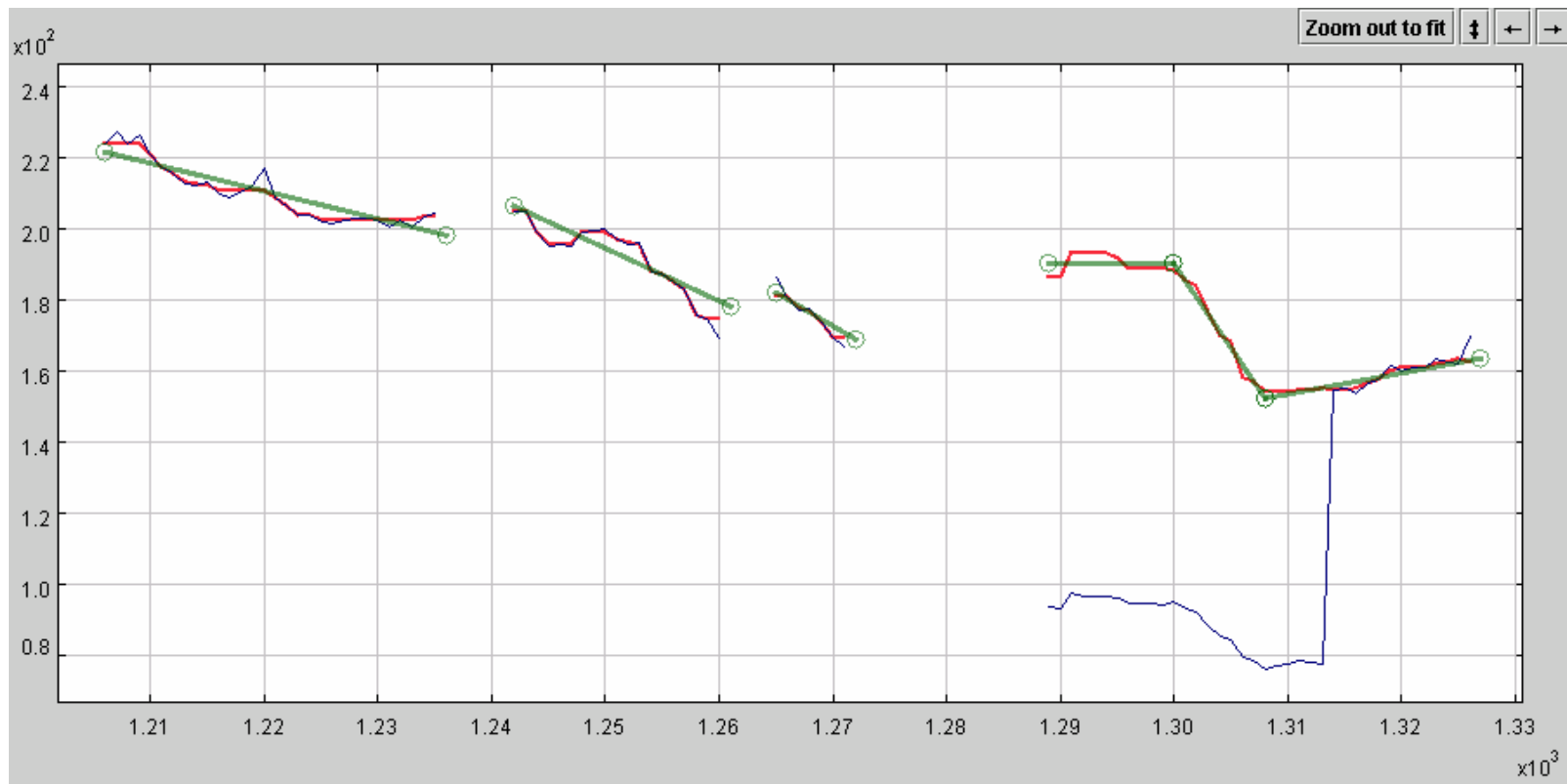
Right Panel: A green panel with the "algeMy" logo, the text "version 0.8 beta", and the names of the developers: Harry Bratt, Federico Cesari, Luciana Ferrer, and Martin Graziarena. It also includes the copyright information: "Copyright 2006 SRI International".

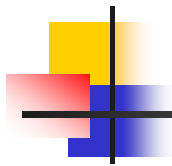
Bottom Panel: A Windows taskbar showing the "start" button and several open applications, including "Excel", "Java...", "TOD...", "Inbo...", "STA...", "kdd...", "cs22...", "supe...", "sre0...", "sre0...", "xterm", "Alge...", "unbt...", "Desktop", "EN", and the system clock showing "4:01 PM".



Prosodic Features

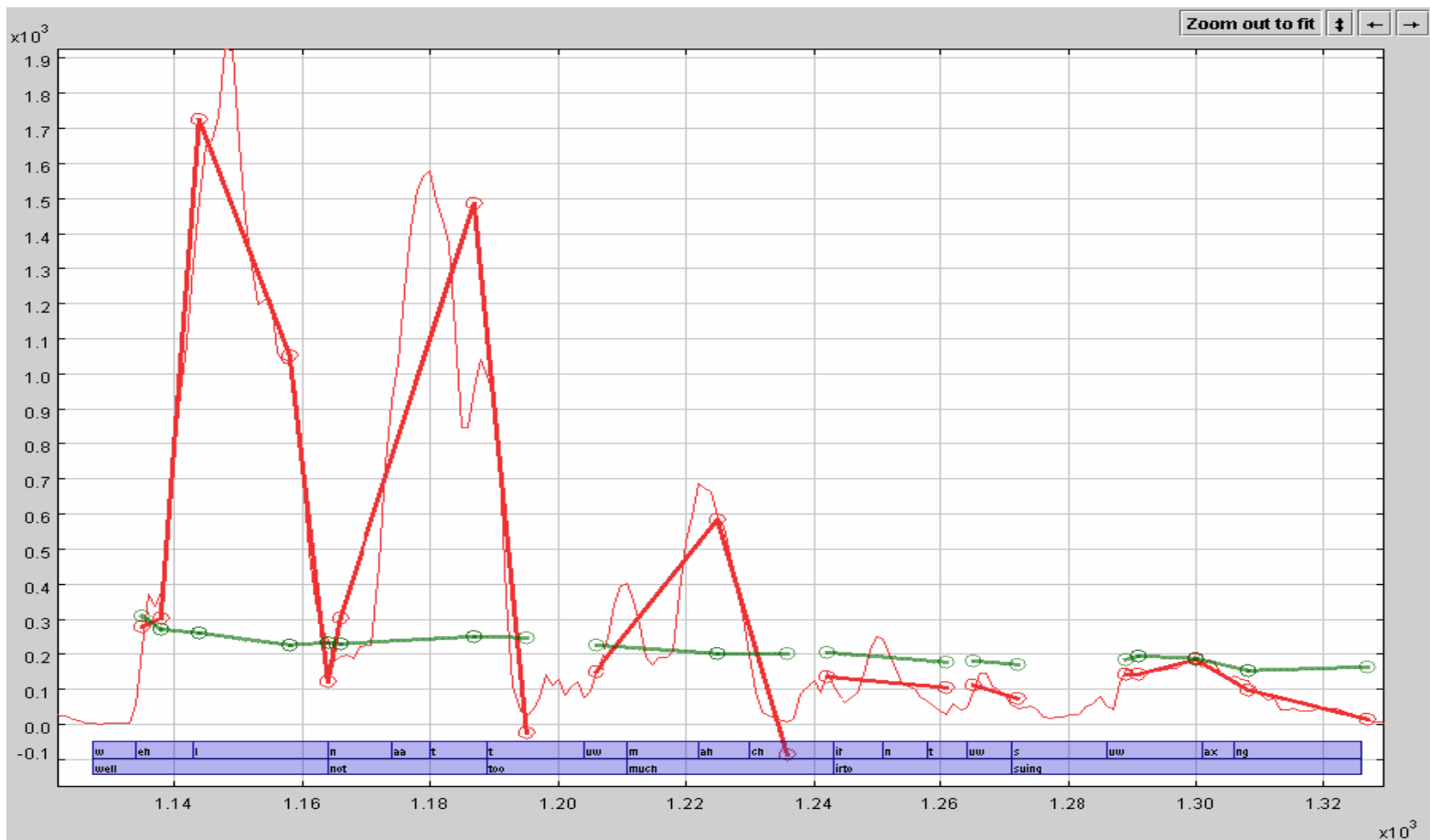
- Pitch stylization
 - Raw pitch is median-filtered and corrected for halving and doubling
 - Best position of nodes is found and linear approximation of the profile is computed (continuous in nodes)





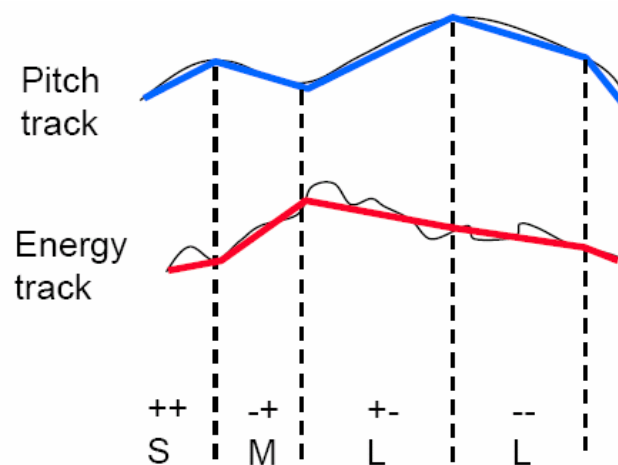
Prosodic Features

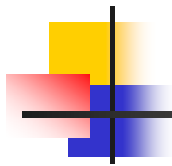
- Energy stylization
 - Based on pitch segmentation
 - Obviously suboptimal, but works well



Prosodic Features: Early approaches

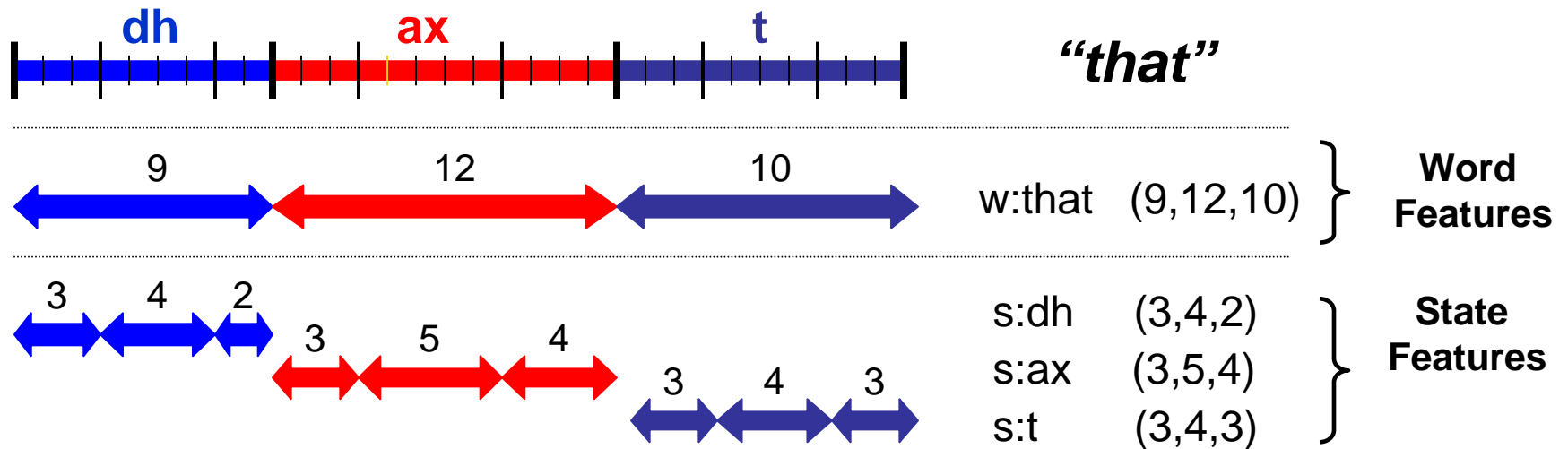
- Simplest approach: Model pitch and energy distributions
 - Features are per-frame $\log(\text{pitch})$, $\log(\text{energy})$ and their deltas for voiced frames
 - Model these features with the UBM-GMM paradigm
- Less simple approach: Model pitch, energy and duration “gestures” (Sonmez'97, Adami'01)
 - Create a sequence of symbols describing the slope of the pitch and energy tracks tagged with quantized duration
 - Use DHMMs to model these features





Duration GMM

- Each word or phone is represented by a feature vector comprised by the durations of the individual phones or states inside it.
- UBM-GMM paradigm is used to model each word or phone.





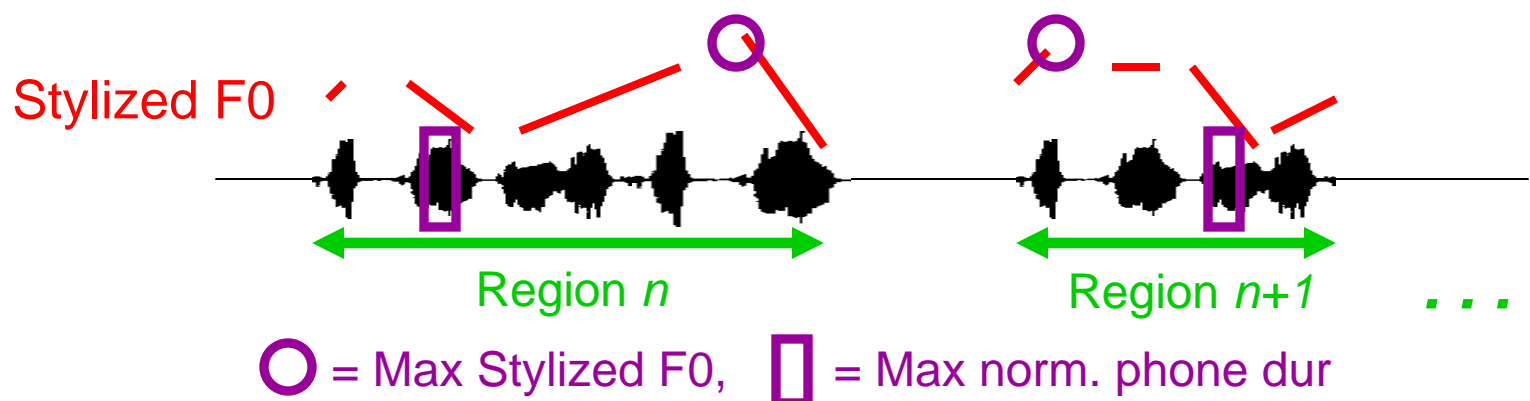
Duration GMM - Results

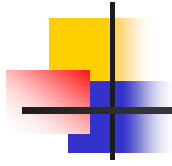
System	1-conv		8-conv	
	% EER	DCF	% EER	DCF
Baseline	7.17	0.248	4.91	0.169
Word Dur	19.22	0.783	8.62	0.374
State Dur	15.36	0.603	8.02	0.319
Baseline + Word Dur	6.10	0.239	3.29	0.111
Baseline + State Dur	6.10	0.211	3.71	0.126
Baseline + Both	5.68	0.205	3.23	0.109

- Word Dur system makes better use of the extra training data than the state dur system

Non-uniform Extraction Region Features (NERFs)

- Idea : use features inside spans of speech where the spans are:
 - Larger than a frame (to capture longer term behaviors)
 - Much smaller than a whole conversation side (to yield enough samples)
- A NERF has 2 ingredients:
 - the “NER” → **region** other than the standard frame-based regions
 - the “F” → the **feature** extracted inside the region
- Regions and features motivated by psycholinguistics & availability
- Example: “pause to pause” (idea units)



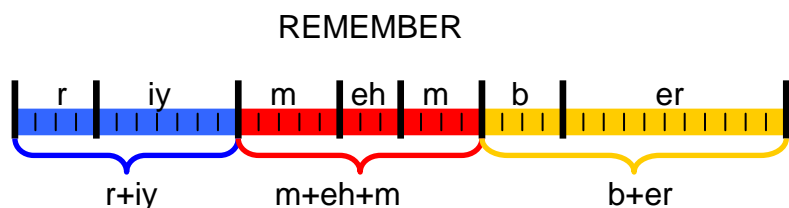


Example Features

- **Pause features:**
 - Length of pauses before and after the region.
- **Duration features:**
 - Total duration of the region.
 - Maximum vowel duration in the region
- **Pitch features:**
 - Mean/max/min stylized pitch in the region.
 - First/last pitch slope in region. Length of that slope.
 - Value of the max positive/negative pitch slope in the region.
- **Energy features:**
 - First/last slope in the stylized energy.
 - Value of the max positive/negative energy slope in the region.

Syllable NERFs

- Syllables are abundant and meaningful as units
- Discretize features so we can model sequences, not just distributions



Compute Syllable-Level Prosodic Features: pitch, energy, duration

Discretize Features

Create N-grams

Obtain Conversation-Level N-gram Frequencies

Prune Feature List

Rank-Norm N-gram Frequencies

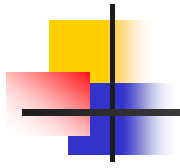
Model Features Using SVM



Syllable NERFs - Results

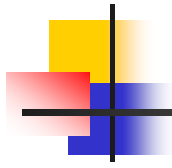
System	1-conv		8-conv	
	% EER	DCF	% EER	DCF
Baseline	7.17	0.248	4.91	0.169
Wnerf+Snerfs	14.06	0.522	6.52	0.274
Baseline + Word SVM	6.42	0.228	3.23	0.123
Baseline + Both Dur	5.68	0.205	3.23	0.109
Baseline + Wnerf+Snerfs	5.91	0.210	3.41	0.103

- Best of all stylistic system individually and very good in combination with baseline
- But really complicated to implement and slow to run (getting better with use of Aljemy)



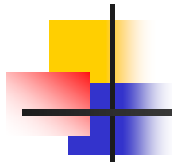
Agenda

- Introduction
- Baseline GMM-UBM system
- Improving the baseline
 - Other cepstral systems
 - Stylistic systems
- **Combination issues**
- Conclusions



Combination issues

- We run all systems separately and combine their scores at the end
 - This is clearly suboptimal, but easiest to do ...
- People have been using a simple no-hidden layer Neural Network for combination
 - Ends up being a linear combination with sophisticated training procedure
- We have tried many many many other things but nothing beats the NN consistently
- Results shown in this talk refer to a combination of combiners that worked well last year
- This would require a whole other lecture ...



Combination Results

Contribution from Acoustic vs. Stylistic Systems

Systems Included	1-conv		8-conv	
	EER	DCF	EER	DCF
Baseline	7.17	0.248	4.91	0.169
Baseline + newAcoustic	4.61	0.166	2.45	0.080
Baseline + Stylistic	4.89	0.177	2.45	0.077
Baseline + newAcoustic + Stylistic	4.10	0.131	1.97	0.054
Overall Relative improvement	43%	47%	60%	68%

- Interestingly, adding acoustic gives about same performance as adding stylistic (acoustic a bit better for 1s, stylistic for 8s)
- Yet combining them gives a large win, in both conditions
- Note: Last line for 8-conv does not include state-dur (see next slide)

Combination Results - Analysis

Best DCF results for N systems

1-conv

8-conv

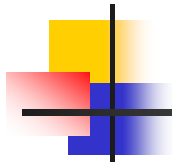
	Ag	Am	Sf	Sw	As	Sn	Ss	DCF	As	Sf	Am	Sn	Sw	Ag	Ss	DCF
1 Best								0.248								0.103
2 Best								0.177								0.074
3 Best								0.158								0.064
4 Best								0.144								0.058
5 Best								0.133								0.055
6 Best								0.131								0.054
7 Best								0.131								0.056

Acoustic Systems (A)		Stylistic Systems (S)	
Ag	Cepstral GMM	Sn	Word N-grams
As	Cepstral SVM	Sw	Word duration
Am	MLLR SVM	Ss	State duration
		Sf	SNERFs+WNERFs



Agenda

- Introduction
- Baseline GMM-UBM system
- Improving the baseline
 - Other cepstral systems
 - Stylistic systems
- Combination issues
- Conclusions



Conclusions

- Cepstral GMM baseline can be hugely improved with the use of other cepstral and stylistic systems
 - The improvement from adding these systems is bigger when more data is available
- Performance of a system alone does not predict (often inversely related to) importance in a larger combination.
- Both acoustic and stylistic features are important
- System importance depends on amount of training data. More training data means:
 - Greater use of SVM cepstral system
 - Greater use of stylistic features, esp. WNERFs+SNERFs and word N-grams



Thanks !