READ ME ACIDA Version 1.0

Veena Singampalli and Philip Jackson

February 16, 2009

Contents

1 Overview

Articulation constraint identification algorithm (ACIDA) is used for identification of critical dimensions from the EMA data from MOCHA-TIMIT database. The current version of the algorithm can be implemented on the data from male (msak0) and female (fsew0) speaker data. ACIDA currently operates on 1D data where x and y movements of articulatory coordinates were treated independently and 2D data where the correlation between x and y movements of articulatory coordinates was considered . The structure of the current release <A0cida_1-0>, the scripts and the implementation details are explained in this file.

2 Directory structure

Folder <Acida_1-0> has the scripts for 1D and 2D versions of the algorithm along with the 1D and 2D input files containing grand and phone statistics generated for male speaker (msak0) from the Mocha database. The main program for obtaining a list of critical dimensions is *icrit_1d.m* for 1D case and *icrit_2d.m* for the 2D case. Scripts *computeIdiv_1d.m* and *updateDep_1d.m* for 1D case (*computeIdiv_2d.m* and *updateDep_2d.m* for 2D case) contain functions for computation of identification divergence and D-step respectively. Scripts *disp1Dres.m* and *disp2Dres.m* display the 1D and 2D critical dimensions for every phone respectively. The input files provided in this tool box for male speaker are `1dstats_m.mat` for 1D and `2dstats_m.mat` for 2D case respectively.

3 ACIDA outline

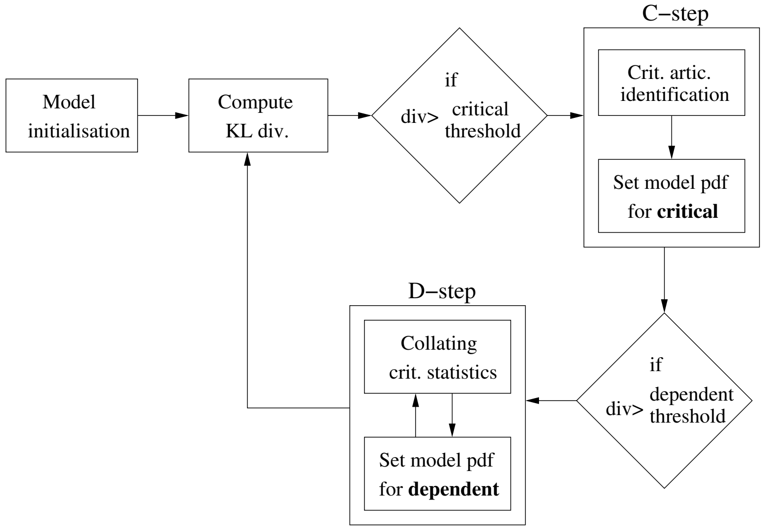

Figure 1: *Data flow illustration of different stages in ACIDA.*

A model distribution defined by a model mean and variance is assigned to every articulatory coordinate for all phones in the data. The model distribution is univariate Gaussian for 1D case and bivariate Gaussian for 2D version of the algorithm. Figure 1 shows the stages and flow of the algorithm for identification of articulatory roles. The algorithm operates in the following four stages

1. Model initialisation: In this stage, the model means and variances of all articulatory coordinates for each phone are set to the grand means and variances. It is assumed in this study that the grand distributions reflect the neutral state configurations of the articulatory coordinates. At this stage, all the articulatory coordinates of the phone defined by the model distribution are in the neutral position.

2. Divergence calculation: The KL divergence between the model distributions and the phone-specific distributions of all articulatory coordinates for each phone is calculated.

3. Critical identification step (C-step): Here, the articulatory dimension associated with the maximum divergence is identified as critical. The model distribution of the critical coordinate is updated by setting it to the phone distribution.

4. Dependent update step (D-step): The dependent coordinates are identified based on the grand articulatory correlations and their distributions are updated.

The model distributions of articulatory coordinates are updated based on the grand articulatory correlations in each iteration. As shown in fig.1, steps 2 to 4 are repeated until the divergence exceeds a threshold value known as the **critical** threshold. For each phone, the algorithm identifies a set of critical articulatory coordinates and estimates the model distributions of articulators according to the role played by them for that phone.

4 Program

The main program for obtaining a list of critical dimensions is *icrit_1d.m* for 1D case and *icrit_2d.m* for the 2D case. The following sections explain the inputs, outputs, options and functions in each main program.

4.1 Options

The following flags present in the main program can be enabled and disabled to do specific tasks

Option	Value	Description
`isMale`	1	male speaker data and results
	0	female speaker data and results
`isSave`	1	save results
	0	don't save results
`isDisp`	1	display results
	0	don't display results

Table 1: Flags in the main program

4.2 Variables

The main program consists of the following variables that can be modified as per requirement

Variable name	value	description
`n`	14	no: critical coordinates for 1D
	7	no: critical coordinates for 1D
`nPhone`	51	no: phones in the database
`thetaC`	any positive rational number	critical threshold
`thetaD`	0.1	dependent threhsold

Table 2: Variables in the main program

4.2.1 Grand and phone statistics

The statistics in tab. 3, stored in data file called `1dstats_m.mat` for male (`1dstats_f.mat` for female) are needed for running 1D ACIDA. For 2D ACIDA, `2dstats_m.mat` and `2dstats_f.mat` are needed. This toolbox comes with the 1D and 2D data required for running the algorithm for male speaker. Input files can be generated in the similar way for female speaker (fsew0). The algorithm can be run on any database by making modifications where necessary. For eg., adjustments to the dimensionality of articulatory space `n` or the number of phones `nPhone` might be necessary for a dataset with different dimensionality or phonetic content.

4.2.2 Articulator and phone lists

- Articulator list for 1D denoted by `artic_label` is [ULx, ULy, LLx, LLy, LIx, LIy, TTx, TTy, TBx, TBy, TDx, TDy, Vx, Vy].
- Articulator list for 2D denoted by `artic_label` is [UL, LL, LI, TT, TB, TD, V].
- Phone list is stored in file "newHmmList" in the form of
  p
  b
  m
  . . .

Input	Notation in pseudocode	Variable name	Size (1D)	Size (2D)
Grand mean	M	`M`	1×14	2×7
Grand covariance	Σ	`Sigma`	1×14	$2 \times 2 \times 7$
Grand correlations	R^*	`R`	14×14	14×14
Grand sample size	N	`N`	1×1	1×1
Phone mean	μ^ϕ	`muPhi`	14×51	$2 \times 7 \times 51$
Phone covariance	Σ^ϕ	`sigmaPhi`	14×51	$2 \times 2 \times 7 \times 51$
Phone correlations	R^ϕ	`Rphi`	$14 \times 14 \times 51$	$14 \times 14 \times 51$
Phone sample sizes	N^ϕ	`Nphi`	1×51	1×51

Table 3: Information on variable names, dimensionality and notation of grand and phone statistics for data from MOCHA database.

4.3 Output

Output	Notation in pseudocode	Variable name	Size (1D)	Size (2D)
Model mean	$\hat{m}^\phi$	`hatm`	1×14	2×7
Model covariance	$\hat{S}^\phi$	`hatS`	1×14	$2 \times 2 \times 7$
Critical coordinate list	$\hat{C}^\phi$	`hatC`	14×14	14×14
No: critical coordinates	K^ϕ	`K`	1×1	1×1

Table 4: Output generated by the program

5 Functions

There are two functions in each program

1. *computeIdiv_1d* for 1D case and *computeIdiv_2d* for 2D case
2. *updateDep_1d* for 1D case and *updateDep_2d* for 2D case

5.1 computeIdiv

- Purpose: computation of identification divergence
- Inputs : Means, covariance and sample sizes of model and phone distributions
- Output: Identification divergence J

5.2 updateDep

- Purpose: computation of dependent update or D-step
- Inputs : Means, covariance and sample sizes of model, grand and phone distributions along with grand and phone correlations
- Output: updated model means M and covariances S