List of modifications made in HTS (for version 2.1)

Heiga ZEN (Byung Ha CHUN)

June 27, 2008

1 Modifications in Model Definition

In HTS, the HTK HMM definition (please see HTKBook [1] Chapter 7) has been modified to support MSD [2], stream-level tying, and adaptation of multi-stream HMMs. This section gives its brief description.

First, <MSDInfo> is added to global options of the HTK HMM definition language The arguments to the <MSDInfo> option are the number of streams (default 1) and then for each stream, 0 (non-MSD stream) or 1 (MSD stream) of that stream. The full set of global options in HTS is given below.

globalOpts	=	option { option }
option	=	<HmmSetId> string |
		<StreamInfo> short { short } |
		<MSDInfo> short { short } |
		<VecSize> short |
		<ProjSize> short |
		<InputXform> inputXform |
		<ParentXform> ∼a macro |
		covkind |
		durkind |
		parmkind

Second, the number of mixture specification is modified to support stream-level tying structure as follows:

```
HTK                            HTS
 <State> 2                      <State> 2
  <NumMixes> 1 2
  <SWeights> 2 0.9 1.1           <SWeights> 2 0.9 1.1
  <Stream> 1                     <Stream> 1
                                    <NumMixes> 1
     <Mixture> 1 1.0                <Mixture> 1 1.0
      <Mean> 4                       <Mean> 4
        0.3 0.2 0.1 0.0                0.3 0.2 0.1 0.0
      <Variance> 4                   <Variance> 4
        0.5 0.4 0.3 0.2                0.5 0.4 0.3 0.2
    <Stream> 2                     <Stream> 2
                                    <NumMixes> 2
     <Mixture> 1 0.4                <Mixture> 1 0.4
      <Mean> 2                       <Mean> 2
        1.0 2.0                        1.0 2.0
      <Variance> 2                   <Variance> 2
        4.0 8.0                        4.0 8.0
     <Mixture> 2 0.6                <Mixture> 2 0.6
      <Mean> 2                       <Mean> 2
        2.0 9.0                        2.0 9.0
      <Variance> 2                  <Variance> 2
        3.0 6.0                        3.0 6.0
```

As you can see, <NumMixes> is moved from state-level to stream-level. This modification enables us to include the number of mixture component in the stream-level macro. Based on this implementation, stream-level macro was added. The various distinct points in the hierarchy of HMM parameters which can be tied in HTS is as follows:

∼s	shared state distribution
∼p	shared stream
∼m	shared Gaussian mixture component
∼u	shared mean vector
∼v	shared diagonal variance vector
∼i	shared inverse full covariance matrix
∼c	shared Cholesky U matrix
∼x	shared arbitrary transform matrix
∼t	shared transition matrix
∼d	shared duration parameters
∼w	shared stream weight vector

Note that the ∼p macro is used by the HMM editor HHEd for building tied mixture systems in the original HTK macro definition.

The resultant state definition of in the modified HTK HMM definition language is as follows:

```
state       =  <State> short stateinfo
stateinfo   =  ∼s macro |
               [ weights ] stream { stream } [ duration ]
macro       =  string
weights     =  ∼w macro | <SWeights> short vector
vector      =  float { float }
stream      =  [ <Stream> short ] streaminfo
streaminfo  =  ∼p macro | [ <Stream> short ] [mixes] (mixture { mixture } | tmixpdf | discpdf)
mixes       =  <NumMixes> short {short}
tmixpdf     =  <TMix> macro weightList
weightList  =  repShort { repShort }
repShort    =  short [ ∗ char ]
discpdf     =  <DProb> weightList
mixture     =  [ <Mixture> short float ] mixpdf
mixpdf      =  ∼m macro | mean cov [ <GConst> float ]
mean        =  ∼u macro | <Mean> short vector
cov         =  var | inv | xform
var         =  ∼v macro | <Variance> short vector
inv         =  ∼i macro |
               (<InvCovar> | <LLTCovar>) short tmatrix
xform       =  ∼x macro | <Xform> short short matrix
matrix      =  float {float}
tmatrix     =  matrix
```

It should be noted that <Stream> can doubly be specified in both stream and streaminfo. This is because <Stream> in ∼p macro is essential to specify stream index of this macro. This stream index information is used in various HTS functions to check stream consistency.

Third, to support multi-stream HMM adaptation, the HTK HMM definition language for baseclasses is modified. A baseclass is defined as

```
baseClass   =  ∼b macro baseopts classes
baseopts    =  <MMFIdMask> string <Parameters> baseKind [<StreamInfo>] <NumClasses> int
StreamInfo  =  short { short } |
baseKind    =  MIXBASE | MEANBASE | COVBASE
classes     =  <Class> int itemlist { classes }
```

where <StreamInfo> is optionally added to specify the stream structure.

2 Added Configuration Variables

A number of configuration variables have been added to HTK to control new functions implemented in HTS. Their names, default values, and brief descriptions are as follows:

Module	Name	Default	Description
HADAPT	SAVEFULLC	F	Save transformed model set in full covariance form
	USESMAP	F	Use structural MAP criterion [3]
	SMAPSIGMA	1.0	Prior parameter for SMAP criterion
	SAVEALLSMAPXFORM	T	Save all (unnecessary) linear transforms estimated in SMAPLR/CSMAPLR
	BANDWIDTH		Bandwidth of transformation matrices [4]
	DURUSEBIAS	F	Specify a bias with linear transforms
	DURSPLITTHRESH	1000.0	Minimum occupancy to generate a transform for state duration model set
	DURTRANSKIND	MLLRMEAN	Transformation kind
	DURBLOCKSIZE	full	Block structure of transform for state duration model set
	DURBANDWIDTH		Bandwidth of transformation matrices for state duration model set
	DURBASECLASS	global	Macroname of baseclass for state duration model set
	DURREGTREE		Macroname of regression tree for state duration model set
	DURADAPTKIND	BASE	Use regression tree or base classes to adapt state duration model set
HFB	MAXSTDDEVCOEF	10	Maximum duration to be evaluated
	MINDUR	5	Minimum duration to be evaluated
HMAP	APPLYVFLOOR	T	Apply variance floor to model set
HGEN	MAXEMITER	20	Maximum # of EM iterations
	EMEPSILON	1.0E-4	Convergence factor for EM iteration
	RNDPARMEAN	0.0	Mean of Gaussian noise for random generation [5]
	RNDPARVAR	1.0	Variance of Gaussian noise for random generation
	USEGV	F	Use speech parameter generation algorithm considering GV [6]

Module	Name	Default	Description
	CDGV	F	Use context-dependent GV model set
	LOGGV	F	Use logarithmic GV instead of linear GV
	MAXGVITER	F	Max iterations in the speech parameter generation considering GV
	GVEPSILON	1.0E-4	Convergence factor for GV iteration
	MINEUCNORM	1.0E-2	Minimum Euclid norm of a gradient vector
	STEPINIT	1.0	Initial step size
	STEPDEC	0.5	Step size deceleration factor
	STEPINC	1.2	Step size acceleration factor
	HMMWEIGHT	1.0	Weight for HMM output prob
	GVWEIGHT	1.0	Weight for GV output prob
	OPTKIND	NEWTON	Optimization method
	RNDFLAGS		Random generation flag
	GVMODELMMF		GV MMF file
	GVHMMLIST		GV model list
	GVMODELDIR		Dir containing GV models
	GVMODELEXT		Ext to be used with above Dir
	GVOFFMODEL		Model names to be excluded from GV calculation
HMODEL	IGNOREVALUE	-1.0E+10	Ignore value to indicate zero-dimensional space in multi-space probability distribution
HCOMPV	NSHOWELEM	12	# of vector elements to be shows
	VFLOORSCALE	0.0	variance flooring scale
	VFLOORSCALESTR		variance flooring scale vector for streams
HEREST	APPLYVFLOOR	T	Apply variance floor to model set
	DURMINVAR	0.0	Minimum variance floor for state duration model set
	DURVARFLOORPERCENTILE	0	Maximum number of Gaussian components (as the percentage of the total Gaussian components in the system) to undergo variance floor for state duration model set
	APPLYDURVARFLOOR	T	Apply variance floor to state duration model set
	DURMAPTAU	0.0	MAP tau for state duration model set [7]
	ALIGNDURMMF		State duration MMF file for alignment (2-model reest)

Module	Name	Default	Description
	ALIGNDURLIST		State duration model list for alignment (2-model reest)
	ALIGNDURDIR		Dir containing state duration models for alignment (2-model reest)
	ALIGNDUREXT		Ext to be used with above Dir (2-model reest)
	ALIGNDURXFORMEXT		Input transform ext for state duration model set to be used with 2-model reest
	ALIGNDURXFORMDIR		Input transform dir for state duration model set to be used with 2-model reest
	DURINXFORMMASK		Input transform mask for state duration model set (default output transform mask)
	DURPAXFORMMASK		Parent transform mask for state duration model set (default output parent mask)
HHED	USEPATTERN	F	Use pattern instead of base phone for tree-based clustering
	SINGLETREE	F	Construct single tree for each state position
	APPLYMDL	F	Use the MDL criterion for tree-based clustering [8]
	IGNORESTRW	F	Ignore stream weight in tree-based clustering
	REDUCEMEM	F	Use reduced memory implementation of tree-based clustering
	MINVAR	1.0E-6	Minimum variance floor for model set
	MDLFACTOR	1.0	Factor to control the model complexity term in the MDL criterion
	MINLEAFOCC	0.0	Minimum occupancy count in each leaf node
	MINMIXOCC	0.0	Minimum occupancy count in each mixture component
	SHRINKOCCTHRESH		Minimum occupancy count in decision trees shrinking
HMGENS	SAVEBINARY	F	Save generated parameters in binary
	OUTPDF	F	Output pdf sequences
	PARMGENTYPE	0	Type of parameter generation algorithm [9]
	MODELALIGN	F	Use model-level alignments given from label files to determine model-level durations

Module	Name	Default	Description
	STATEALIGN	F	Use state-level alignments given from label files to determine state-level durations
	USEALIGN	F	Use model-level alignments to prune EM-based parameter generation algorithm
	USEHMMFB	F	Do not use state duration models in the EM-based parameter generation algorithm
	INXFORMMASK		Input transform mask
	PAXFORMMASK		Parent transform mask
	PDFSTRSIZE		# of PdfStreams
	PDFSTRORDER		Size of static feature in each PdfStream
	PDFSTREXT		Ext to be used for generated parameters from each PdfStream
	WINEXT		Ext to be used for window coefficients file
	WINDIR		Dir containing window coefficient files
	WINFN		Name of window coefficient files

Other configuration variables in HTK can also be used with HTS. Please refer to HTKBook [1] Chapter 18 for others.

3 Added Command-Line Options

Various new command-line options have also been added to HTK tools. They are listed as follows:

HINIT

```
Option                                                      Default
 -g      Ignore outlier vector in MSD                        on
```

HREST

```
Option                                                      Default
 -g s    output duration model to file s                     none
 -o fn   Store new hmm def in fn (name only)                 outDir/srcfn
```

HEREST

```
Option                                                      Default
 -b      use an input linear transform for dur models        off
 -f s    extension for new duration model files              as src
 -g s    output duration model to file s                     none
 -n s    dir to find duration model definitions              current
 -q s    save all xforms for duration to TMF file s          TMF
 -u tmvwapd update t)rans m)eans v)ars w)ghts                tmvw
                   a)daptation xform p)rior used
                   s)semi-tied xform
                   d) switch to duration model update flag
 -y s    extension for duration model files                  none
 -N mmf  load duration macro file mmf
 -R dir  dir to write duration macro files                   current
 -W s [s] set dir for duration parent xform to s             off
         and optional extension
 -Y s [s] set dir for duration input xform to s              none
         and optional extension
 -Z s [s] set dir for duration output xform to s             none
```

HHED

```
Option                                                      Default
 -a f    factor to control the second term in the MDL        1.0
 -i      ignore stream weight                                off
 -m      apply MDL principle for clustering                  off
 -p      use pattern instead of base phone                   off
 -r      reduce memory usage on clustering                   off
 -s      construct single tree                               off
 -v f    Set minimum variance to f                           1.0E-6
```

HMGENS

```
Option                                                      Default
```

```
  -a      Use an input linear transform for HMMs          off
  -b      Use an input linear transform for dur models     off
  -c n    type of parameter generation algorithm          0
           0: both mix and state sequences are given
           1: state sequence is given,
              but mix sequence is hidden
           2: both state and mix sequences are hidden
  -d s    dir to find hmm definitions                     current
  -e      use model alignment from label for pruning      off
  -f f    frame shift in 100 ns                           50000
  -g f    Mixture pruning threshold                       10.0
  -h s [s] set speaker name pattern to s,                 *.%%%
          optionally set parent patterns
  -m      use model alignment for duration                off
  -n s    dir to find duration model definitions          current
  -p      output pdf sequences                            off
  -r f    speaking rate factor (f<1: fast  f>1: slow)     1.0
  -s      use state alignment for duration                off
  -t f [i l] set pruning to f [inc limit]                 inf
  -v f    threshold for switching spaces for MSD          0.5
  -x s    extension for hmm files                         none
  -y s    extension for duration model files              none
  -E s [s] set dir for parent xform to s                  off
          and optional extension
  -G fmt  Set source label format to fmt                  as config
  -H mmf  Load HMM macro file mmf
  -I mlf  Load master label file mlf
  -J s [s] set dir for input xform to s                   none
          and optional extension
  -L dir  Set input label (or net) dir                    current
  -M dir  Dir to write HMM macro files                    current
  -N mmf  Load duration macro file mmf
  -S f    Set script file to f                            none
  -T N    Set trace flags to N                            0
  -V      Print version information                       off
  -W s [s] set dir for duration parent xform to s         off
          and optional extension
  -X ext  Set input label (or net) file ext               lab
  -Y s [s] set dir for duration input xform to s          none
          and optional extension
```

Please also refer to HTKBook [1] Chapter 17 for other command-line options.

4 Added Commands and Modifications in HHED

Some HHED commands have been added in HTS. They are as follows:

```
AX filename          - Set the Adapt XForm to filename
CM directory         - Convert models to pdf for speech synthesizer
CT directory         - Convert trees/questions for speech synthesizer
DM type macroname    - Delete macro from model-set
DR id                - Convert decision trees to a regression tree
DV                   - Convert full covariance to diagonal variances
IT filename          - Clustering while imposing loaded tree structure
                       If any empty leaf nodes exist, loaded trees
                       are pruned
                       and then saved to filename
IX filename          - Set the Input Xform to filename
JM hmmFile itemlist  - Join Models on stream or state level
PX filename          - Set the Parent Xform to filename
// comment           - Comment line (ignored)
```

In many HHED commands, we are required to specify item lists to specify a set of items to be processed. In HTS, item list specification has been modified to specify stream-level items.

```
itemList   =  “{” itemSet { “,” itemSet } “}”
itemSet    =  hmmName . [“transP” | “state” state ]
hmmName    =  ident | identList
identList  =  “(” ident { “,” ident } “)”
ident      =  < char | metachar >
metachar   =  “?” | “⋆”
state      =  index [“.” stateComp ]
index      =  “[” intRange { “,” intRange } “]”
intRange   =  integer [ “-” integer ]
stateComp  =  “dur” | “weights” | stream
stream     =  [ “ stream” index ] [ “.mix” mix ]
mix        =  index [ “.” ( “mean” | “cov” ) ]
```

For example,

```
TI str1 {*.state[2].stream[1]}
```

denotes tying streams in state 2 of all phonemes.

Appendix A History of HTS

- **Version 1.0 (December 2002)**
  - Based on HTK-3.2.
  - HHED supports tree-based clustering based on the MDL criterion [8].
  - HHED supports stream-dependent tree-based clustering [10].
  - HMODEL supports multi-space probability distributions (MSD) [2].
  - HEREST can generate state duration modeling [11].
  - Speech parameter generation algorithm [9] is implemented in HGEN and HMGENS.
  - Demo using the CMU Communicator database.
- **Version 1.1 (May 2003)**
  - Based on HTK-3.2.
  - Small run-time synthesis engine (hts_engine).
  - Demo using the CSTR TIMIT database.
  - HTS voices for the Festival speech synthesis system [12].
- **Version 1.1.1 (December 2003)**
  - Based on HTK-3.2.1.
  - HCOMPV supports variance flooring for MSD-HMMs.
  - Demo using the CMU ARCTIC database [13].
  - Demo using the Nitech Japanese database.
  - Demo supports post-filtering [14].
  - HTS voice for the Galatea toolkit [15].
- **Version 2.0 (December 2006) [16]**
  - Based on HTK-3.4.
  - Support generating state duration PDFs in HREST.
  - Phoneme boundaries can be given to HEREST using the -e option [17].
  - Reduced-memory implementation of tree-based clustering in HHED with the -r option.
  - Each decision tree can have a name with regular expressions in HHED with the -p option.
  - Flexible model structures in HMGENS.
  - Speech parameter generation algorithm based on the EM algorithm [9] in HMGenS.
  - Random generation algorithm [5] in HMGENS [5].
  - State or phoneme-level alignments can be given to HMGENS.
  - The interface of HMGENS has been switched to HEREST-style.
  - Various kinds of linear transformations for MSD-HMMs are supported in HADAPT.
    * Constrained MLLR based adaptation [18].
    * Adaptive training based on constrained MLLR [18].
    * Precision matrix modeling based on semi-tied covariance matrices [19].
    * Heteroscedastic linear discriminant analysis (HLDA) based feature transform [20].
    * Phonetic decision trees can be used to define regression classes for adaptation [21]
    * Adapted HMMs can be converted to the run-time synthesis engine format.
  - Maximum a posteriori (MAP) adaptation [7] for MSD-HMMs in HMAP.
- **Version 2.0.1 (May 2007)**
  - Based on HTK-3.4.
  - HADAPT supports band structure for linear transforms [4].
  - HCOMPV supports stream-dependent variance flooring scales.
  - Demo scripts support LSP-type spectral parameters.
  - β version of the runtime synthesis engine API (hts_engine API).
  - hts_engine API supports speaker interpolation [22].
- **Version 2.1 (June 2008)**
  - Based on HTK-3.4.
  - Released under the New and Simplified BSD license [23].

– Simple documentation.
– 64-bit compile support.
– MAXSTRLEN (maximum length of strings), MAXFNAMELEN (maximum length of filenames), PAT_LEN (maximum length of pattern strings), and SMAX (maximum number of streams) defined in HShell.h can be set through configure script.
– HFB supports the forward/backward algorithm for hidden semi-Markov models (HSMMs) [24, 25].
– HADAPT suports SMAPLR/CSMAPLR adaptation [26, 3].
– HGEN supports speech parameter generation algorithm considering global variance (GV) [6].
– HGEN supports random generation of transitions, durations, and mixture components.
– HEREST supports HSMM training and adaptation.
– HMGENS supports speech parameter generation from HSMMs.
– Add DM command to HHED to delete an existing macro from MMF.
– Add IT command to HHED to impose pre-constructed trees in clustering.
– Add JM command to HHED to join models on state or stream level.
– HHED MU command supports '*2' style mixing up.
– HHED MU command supports mixture-level occcupancy threshhold in mixing up.
– First stable version of the runtime synthesis engine API (hts_engine API).

References

[1] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X.-Y. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The Hidden Markov Model Toolkit (HTK) version 3.4*, 2006. http://htk.eng.cam.ac.uk/.

[2] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Multi-space probability distribution HMM. *IEICE Trans. Inf. & Syst.*, E85-D(3):455–464, Mar. 2002.

[3] Y. Nakano, M. Tachibana, J. Yamagishi, and T. Kobayashi. Constrained structural maximum a posteriori linear regression for average-voice-based speech synthesis. In *Proc. Interspeech*, pages 2286–2289, 2006.

[4] L. Qin, Y.-J. Wu, Z.-H. Ling, and R.-H. Wang. Improving the performance of HMM-based voice conversion using context clustering decision tree and appropriate regression matrix. In *Proc. of Interspeech (ICSLP)*, pages 2250–2253, 2006.

[5] K. Tokuda, H. Zen, and T. Kitamura. Reformulating the HMM as a trajectory model. In *Proc. Beyond HMM – Workshop on statistical modeling approach for speech recognition*, 2004.

[6] T. Toda and K. Tokuda. A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. *IEICE Trans. Inf. & Syst.*, E90-D(5):816–824, 2007.

[7] J.L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. on Speech & Audio Process.*, 2(2):291–298, 1994.

[8] K. Shinoda and T. Watanabe. MDL-based context-dependent subword modeling for speech recognition. *J. Acoust. Soc. Jpn.(E)*, 21(2):79–86, 2000.

[9] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proc. ICASSP*, pages 1315–1318, 2000.

[10] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proc. Eurospeech*, pages 2347–2350, 1999.

[11] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Duration modeling for HMM-based speech synthesis. In *Proc. ICSLP*, pages 29–32, 1998.

[12] A.W. Black, P. Taylor, and R. Caley. The festival speech synthesis system. http://www.festvox.org/festival/.

[13] J. Kominek and A.W. Black. CMU ARCTIC databases for speech synthesis. Technical Report CMU-LTI-03-177, Carnegie Mellon University, 2003.

[14] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Incorporation of mixed excitation model and postfilter into HMM-based text-to-speech synthesis. *IEICE Trans. Inf. & Syst. (Japanese Edition)*, J87-D-II(8):1563–1571, Aug. 2004.

[15] Galatea – An open-source toolkit for anthropomorphic spoken dialogue agent. http://hil.t.u-tokyo.ac.jp/galatea/.

[16] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A.W. Black, and K. Tokuda. The HMM-based speech synthesis system version 2.0. In *Proc. ISCA SSW6*, pages 294–299, 2007.

[17] D. Huggins-Daines and A. Rudnicky. A constrained Baum-Welch algorithm for improved phoneme segmentation and efficient training. In *Proc. of Interspeech*, pages 1205–1208, 2006.

[18] M.J.F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech & Language*, 12(2):75–98, 1998.

[19] M.J.F. Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 7(3):272–281, 1999.

[20] M.J.F. Gales. Maximum likelihood multiple projection schemes for hidden Markov models. *IEEE Trans. Speech & Audio Process.*, 10(2):37–47, 2002.

[21] J. Yamagishi, M. Tachibana, T. Masuko, and T. Kobayashi. Speaking style adaptation using context clustering decision tree for HMM-based speech synthesis. In *Proc. ICASSP*, pages 5–8, 2004.

[22] T. Yoshimura, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura. Speaker interpolation for HMM-based speech synthesis system. *J. Acoust. Soc. Jpn. (E)*, 21(4):199–206, 2000.

[23] http://www.opensource.org/licenses/category.

[24] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. A hidden semi-Markov model-based speech synthesis system. *IEICE Trans. Inf. & Syst.*, E90-D(5):825–834, 2007.

[25] J. Yamagishi and T. Kobayashi. Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training. *IEICE Trans. Inf. & Syst.*, E90-D(2):533–543, 2007.

[26] O. Shiohan, Y. Myrvoll, and C.-H. Lee. Structural maximum a posteriori linear regression for fast HMM adaptation. *Computer Speech & Language*, 16(3):5–24, 2002.