Draft Nov 7 2014

Speech synthesis assignment for S-89.5150

Task

Parametrise speech waveforms for one speaker. Build a triphone (or more complex if desired) HMM-model for vocoder parameters and duration following the HTK tutorial (with some tweaks). Synthesise vocoder parameters for a previously unseen sentence. Generate the speech waveform for that sentence.

Goals

1. Pass the course
2. Gain understanding on the HMM as a generative model
3. Apply and practise problem solving skills with speech data in various forms

Expected reporting

1. The usual stuff as required by the professor
2. Include samples of the synthetic speech
3. Some feedback on the time spent on assignment!

Reading

1. Tokuda, Keiichi, et al. "An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features." (1995).
2. Tokuda, Keiichi, et al. “Speech Synthesis Based on Hidden Markov Models.” Proceedings of the IEEE | Vol. 101, No. 5, May 2013
3. HTKBook
4. HTS document
5. SPTK reference

Data

American English speakers slt (female) or bdl (male). Choose one. For example by flipping a coin. Wav files are provided in **/work/courses/T/S/89/5150/general/synthesis/wav**. The data has been divided to train and test sets. Mono-, tri- and quinphone as well as full context labels are provided for both sets in **labels** directory.

Required software

The following software will be provided in the course directory **/work/courses/T/S/89/5150/general/synthesis**:

- HTK patched with HTS

- SPTK (speech processing toolkit)
- Hhead (simplifies feature packaging)

Suggested workflow

1. Get comfortable with vocoding
    - Use SPTK tools to extract vocoder parameters. Use the extracted parameters to regenerate the speech waveform.
    - Compare different vocoders: MCEP, LPC, LSF. Play around a little with dimensionality. Get comfortable with frame-based representation of feature vectors. Select a vocoder that you feel you understand at least a little.
    - Tips:
        - HMM-based speech synthesis at 16kHz typically uses 25ms frames with 5ms frame step.
        - SPTK requires wav files to be converted to raw files first (wav2raw, sox). The encoding must be 4-bit float.
        - x2x command is useful for converting data from one numeric encoding to another, or to get an ascii representation (for debugging…). Note that you can order the ascii output into columns to make frame-by-frame investigation easier
        - Feature extraction mostly starts with framing and windowing (these should be familiar from the course?)
        - Try plotting some frames: The vocoder parameters and their spectral envelopes. Matlab (with signal processing toolkit?) has tools to create spectral envelopes for the different parametric representations. An understanding of the vocoder (parameter ranges and associated spectral shape of vowel frames etc) will be very useful when debugging the synthetic versions!

2. Feature generation
    - Once you've selected a feature type, build the features for all the waveforms.
    - Delta features should be familiar from the course

- You need to combine the spectral components and pitch components for each frame. A sample frame (from [1]):

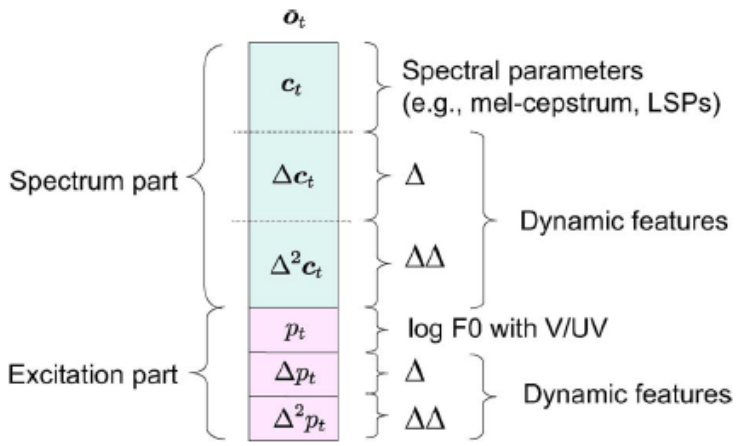

Fig. 3. ***Example of an observation vector at each frame.***

- Create the header for each file. From [3]:

> The HTK file format header is 12 bytes long and contains the following data
> 1. nSamples number of samples in file (4-byte integer)
> 2. sampPeriod sample period in 100ns units (4-byte integer)
> 3. sampSize number of bytes per sample (2-byte integer)
> 4. parmKind a code indicating the sample kind (2-byte integer)

- The numbers can be computed and added to the beginning of the file with SPTK tools, but HHead is provided to simplify the operation; Parameter type is "9" (for "other")
- Tips:
  - SPTK tools delta and bcp (block copy)
  - The delta window needs to match those used in the synthesis.
  - Pitch/F0 is usually converted to log domain where it can be better modelled
  - Pitch/F0 is either 0 (unvoices) or a positive float (voiced); Use "Magic number" definition in delta command for pitch to avoid computing delta features for unvoiced frames.

3. Model Training
   - Follow the tutorial in HTK book. Monophone, triphone, quinphone and full context labels will be provided, as well as the associated list of phonetic questions for clustering. Go as far as you like in model detail!
   - A list of modifications to the training procedure will be provided later
   - Tips:
     - Start by building a monophone model and try to synthesise with it. When it works, do the triphone expansion and synthesise with the

expanded model set. If you feel brave, you can go to even more detailed context of quinphones or full-context, but this is not required.
        - A 5-state proto HMM that uses separate streams for pitch/F0 modelling has been provided in course directory.

4. Synthesis
    - Use HMGenS tool to synthesise vocoder parameters for some labels. Use SPTK tools to generate the speech waveform.
    - Tips:
        - There are a gazillion parameters to tweak. Don't be scared if the first output sounds a little funny, but try to see if the problems are in HMM synthesis or vocoder synthesis. Here is helps if you selected a vocoder whose parameters you understand and can plot some generated frames.
        - Check the output format of HMGenS, make sure to convert to 4-bit float if necessary
        - As you should be using a log-scale pitch/F0, remember to exponentiate it before using it as excitation for vocoder synthesis!

Help!

You are the guinea pigs for this assignment so you might face some challenges that I have not noticed. I do not have fixed office hours, so it is best to email or phone me to arrange meetings in advance. Feel free to email me about problems, I can probably provide quick answers to many of the common problems.

Happy working,

Reima
reima.karhila@aalto.fi
+358 50 430 3384