Lab 2: Αναγνώριση φωνής με το Kaldi toolkit

3 Δεκεμβρίου 2018

Kaldi: Γενικές πληροφορίες

- Εργαλείο για αναγνώριση φωνής (πρόσφατα και άλλα tasks)
- Γραμμένο σε C++
- Εκτενής χρήση FSTs (επεκτείνει τη βιβλιοθήκη OpenFST)
- Πολλές έτοιμες διαδικασίες για γνωστά datasets: Wall Street Journal, LibriSpeech, Tedlium, SwitchBoard, TIMIT

- Άλλες λύσεις για αναγνώριση φωνής: HTK, CMUSphinx
- Το εγχειρίδιο του HTK (HTKBook) περιέχει πολλές πληροφορίες για αναγνώριση φωνής ανεξάρτητες του λογισμικού

Kaldi: Βασικά scripts

- Διαδικασία πρότυπο για όλες τις υπόλοιπες: **wsj**
  - path.sh : συμπεριλαμβάνει όλα τα C++ scripts του Kaldi στο PATH
  - cmd.sh : ορίζει πού θα τρέχουν τα bash scripts της διαδικασίας (cpu, gpu, cluster). Θέτω πάντα όλες του τις μεταβλητές σε **run.pl**
  - φάκελος steps : περιέχει βασικά bash scripts που τρέχουμε κατά τη διαδικασία παραγωγής των μοντέλων
  - φάκελος utils: βοηθητικά scripts (bash, perl, etc.)

Kaldi: Βασικά scripts

- Πώς τρέχω ένα C++ script?
  - Σε ένα terminal: source path.sh ή . ./path.sh
- Βασικοί τύποι αρχείων Kaldi:
  - .ark: binary αρχείο που περιέχει δεδομένα (π.χ. MFCCs)
  - .scp: text αρχείο-δείκτης. Δείχνει πού περιέχονται τα δεδομένα για κάθε πρόταση μέσα στο αντίστοιχο αρχείο .ark
- Πώς δουλεύουν τα C++ scripts που χειρίζονται αρχεία Kaldi?
  - feat-to-dim **ark:**my_ark_file.ark **ark:**output.ark
  - feat-to-dim **ark:**my_ark_file.ark **ark,t:**output.txt
  - feat-to-dim **ark:**my_ark_file.ark **ark,t:**- (αν βάλω παύλα τυπώνει στο stdout)
  - feat-to-dim **scp:**my_scp_file.scp **ark,t:**-

Προπαρασκευή για USC-TIMIT

- 4 ομιλητές (2 άντρες, 2 γυναίκες)
- 1835 προτάσεις συνολικά
- Task: Αναγνώριση φωνημάτων (phoneme recognition)

- Προπαρασκευή:
  - wav.scp
  - text
  - utt2spk
  - spk2utt (με το utils/utt2spk_to_spk2utt.pl)

Kaldi: Γράφος HCLG

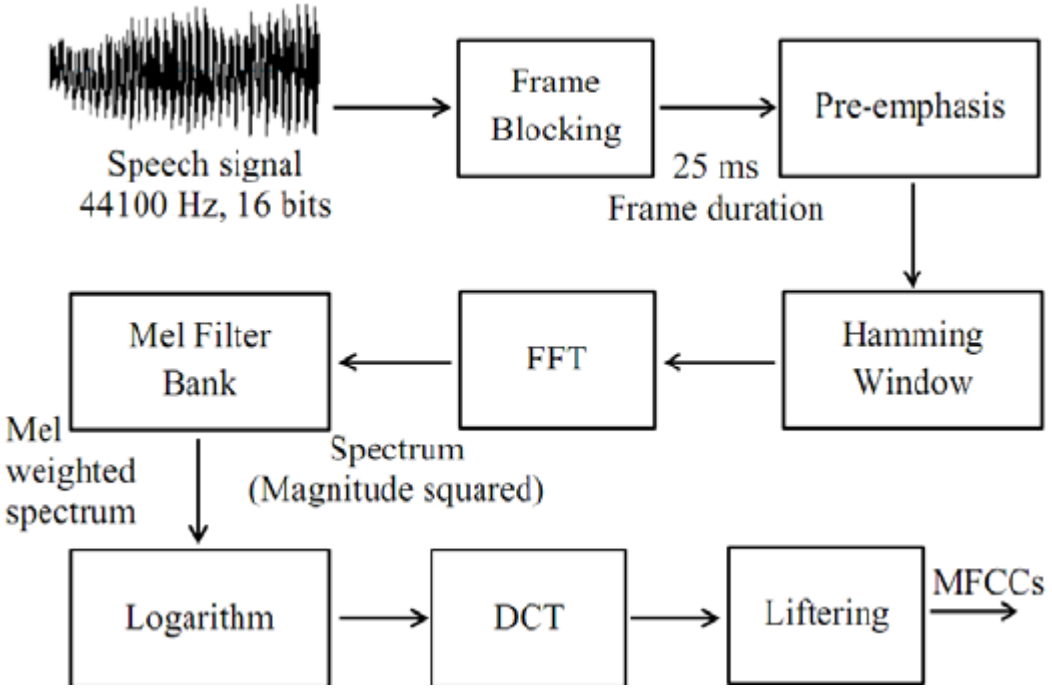
- Σύνθεση 4 γράφων
  - Grammar
  - Lexicon
  - Context dependency
  - Hidden Markov Model
- Για τη δημιουργία του χρειαζόμαστε:
  - Γλωσσικό μοντέλο
  - Ακουστικό μοντέλο
- FSTs που χρειάζεται να δημιουργήσουμε εμείς με το IRSTLM:
  - Grammar: G.fst
  - Lexicon: L.fst

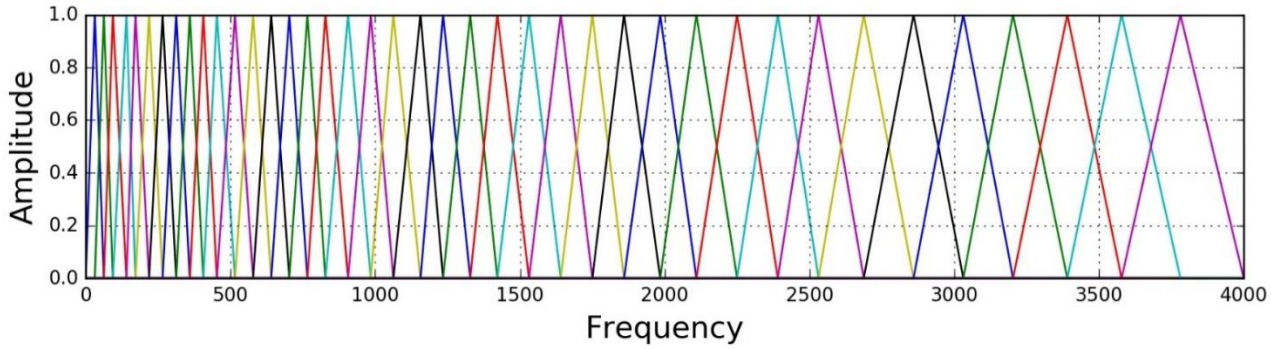
Προετοιμασία γλωσσικού μοντέλου

- Περιεχόμενα data/local/dict:
  - silence_phones.txt
  - optional_silence.txt
  - nonsilence_phones.txt
  - extra_questions.txt
  - lexicon.txt (**Προσοχή:** λεξικό φωνημάτων, διαφορετικό από της προπαρασκευής)
  - lm_train.text
- Εντολές:
  - build-lm.sh
  - compile-lm
  - prepare_lang.sh

Εξαγωγή ακουστικών χαρακτηριστικών MFCCs

- Configuration file: mfcc.conf
- Εντολές: make_mfcc.sh, compute_cmvn_stats.sh





$$f_c^j = 2595 \cdot \log\left(1 + \frac{f^j}{700}\right), j = 1 \dots Q$$

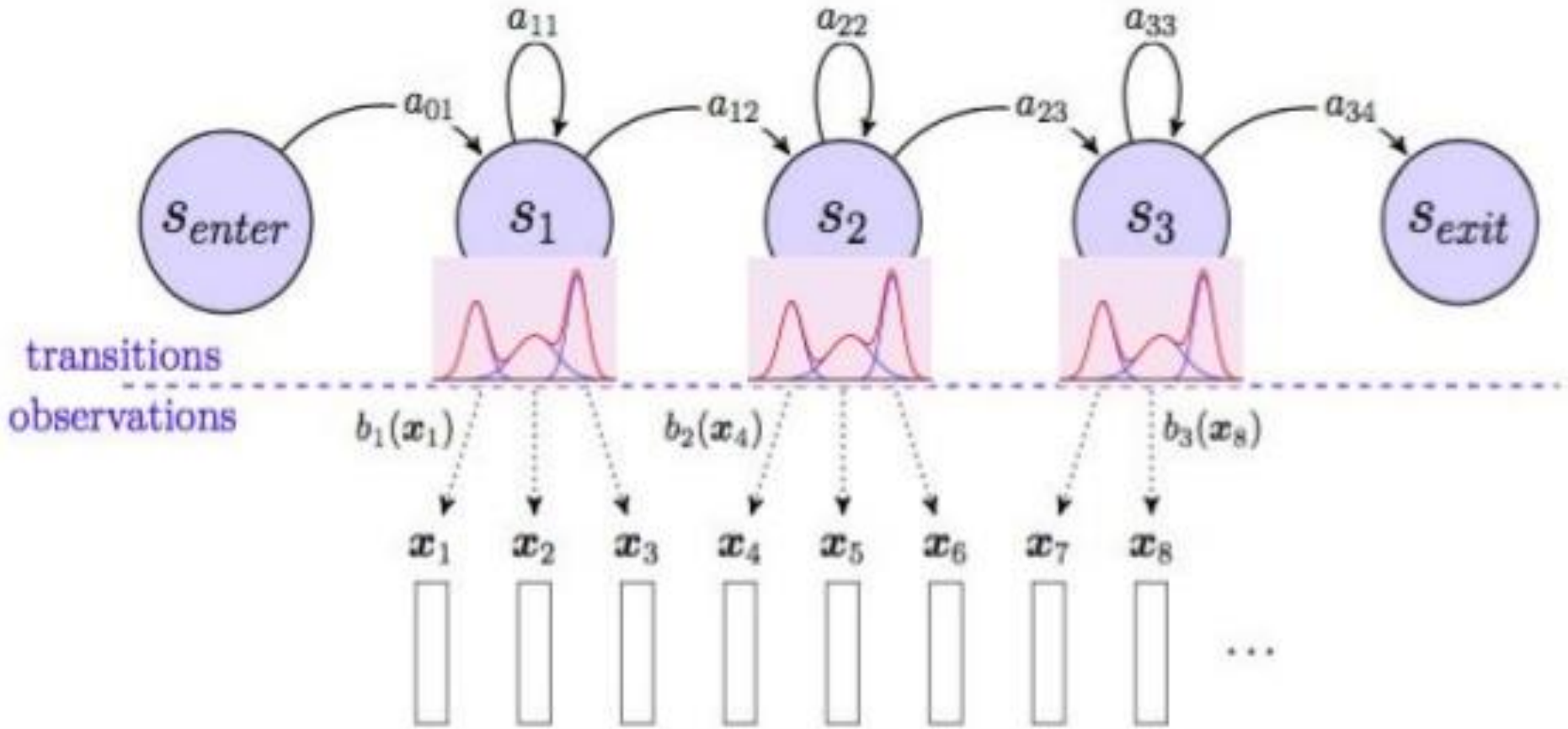
Συστοιχία φίλτρων κλίμακας Mel

Εκπαίδευση ακουστικού μοντέλου – Alignment - Decoding

- Εκπαίδευση ακουστικών μοντέλων:
  - steps/train_mono.sh
  - steps/train_deltas.sh
  - (steps/train_lda_mllt.sh)
- Alignment:
  - steps/align_si.sh
- Decoding:
  - steps/decode.sh

Ακουστικά μοντέλα

- monophone model (δεν περιέχει πληροφορία για προηγούμενα ή επόμενα φωνήματα)
- triphone model (περιέχει πληροφορία γειτονικών φωνημάτων)
  - Εκθετικά μεγαλύτερο σε μέγεθος
  - Phonetic decision tree: δέντρο το οποίο μειώνει το πλήθος όλων των δυνατών καταστάσεων αποκλείοντας αδύνατους συνδυασμούς γειτνίασης
- LDA-MLLT (Linear Discriminant Analysis – Maximum Likelihood Linear Transformation)
  - Μείωση διαστάσεων των χαρακτηριστικών με LDA
  - Εύρεση μοναδικού μετασχηματισμού για κάθε ομιλητή
  - Μειώνει τη διαφορά ανάμεσα στους ομιλητές



Ακουστικό μοντέλο GMM-HMM