MPhil in Machine Learning and Machine Intelligence

Module MLMI2: Speech Recognition

L6: Acoustic Modelling for Large Vocabulary Recognition

Phil Woodland
pcw@eng.cam.ac.uk

Michaelmas 2021

Cambridge University Engineering Department

Outline

This lecture will examine suitable acoustic modelling strategies for large vocabulary speech recognition. We will examine

- the types of units used
- phone units vs word units
- variability of phone realisation
- context dependent phone units
- typical model structures

Context-dependent units can only be used if the number of parameters to be estimated can be constrained. We will examine

- strategies to cluster context-dependent models
- top-down-decision tree state tying

Procedures & results for building an HMM system will be discussed next.

Finally we will introduce the MLMI2 practical.

- HTK Toolkit
- Phone recognition using the TIMIT database

Acoustic Modelling Units

What should be used as the basic units to model speech?

1. Compact (even for large vocabulary tasks);
2. Simple mapping between recognition units and words (sentences);
3. Account for speech variability (e.g. linguistic structure, **co-articulation** between neighbouring speech sounds);
4. Extendable to handle words not seen in the training data;
5. Well-defined and easily identifiable, so that large training corpora may be constructed.

- The choice of unit should also be flexible enough so that the "correct" number of models can be chosen for the available training data while giving good modelling accuracy
- Needs to avoid **over-training** (or over-fitting). Need to balance the number of parameters to the available training data to get optimal performance on test data.

Words as the unit for speech recognition are impractical as:

1. "Too many" models, memory and computation increases as vocab size;
2. Require "vast" amounts of training data to "correctly" estimate the parameters
3. Cannot construct models for words not seen in training data;
4. Cannot capture inter-word co-articulation effects (though intra-word variability very well modelled).

Possible Speech Units

Possible units that have been proposed are:

Phones	40-50 phones in English Highly context dependent Well defined Non-unique phone-string to word-string e.g. *grey twine* and *great wine*
Syllables	10000+ syllables in English Hard to obtain good estimates
Demi-Syllables	2000+ in English, Hard to define

For example the word "Segmentation" may be written as

Phone	/ **s eh g m ax n t ey sh ax n** /
Syllable	/ **seg men ta tion** /

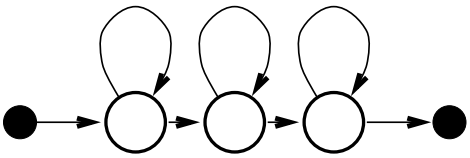
For demi-syllables, split in the vowel in each syllable.

The syllable structure of words can influence the phonetic realisation. However, to date, LVR systems for English using syllable based units are rarely used (some languages such as Mandarin Chinese are more usual to model with syllable-aware units).

Model Structure/Pronunciations

Often the same model structure is used for each basic phone HMM

- Three emitting states;
- Left-to-right structure no skips;



Standard Phone Model

Word pronunciation variability is handled by using multiple pronunciation dictionaries. eg

the = / **dh ax** /
= / **dh iy** /

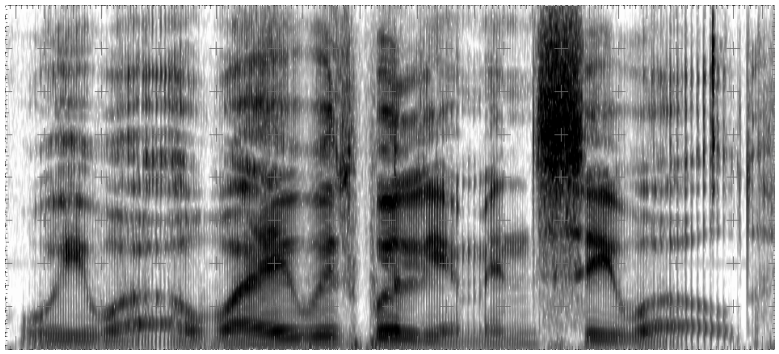
Models typically trained by selecting one of the possible pronunciations as "correct" given the current model set and training using this single pronunciation. An alternative is to make a network of possible pronunciations.

Note that Maximum Likelihood estimation can **overfit** and this includes the parameters of a GMM: if a single observation vector assigned to a component the variances will be estimated as zero. Could use some type of Bayesian estimation (but need prior distribution).

- One simple approach for GMMs with components with diagonal covariance matrices is to have a **variance floor** which is used to set the minimum value of each vector element.
- Could have variable number of GMM components per state (depend on state-occupancy).

Phone Variation

Co-articulation causes the acoustic realisation in a particular context to be more consistent than the same phone occurring in a wide variety of contexts.



The diagram shows a spectrogram for the phrase

We were away with William in Sea World

Each realisation of the *w* phone varies considerably but the most similar are the two occurrences in the same triphone context (underlined).

To handle this problem **context dependent** phone models may be used.

Context Dependent Phones

Context Independent (CI) phone models (**Monophones**) can be made more "specific" by taking into account phonetic context to form **Context Dependent** (CD) models.

The phonetic transcription for the word "Speech" can be written as:

Monophone	/ **s p iy ch** /
Biphones (L)	/ **sil-s s-p p-iy iy-ch** /
Biphones (R)	/ **s+p p+iy iy+ch ch+sil** /
Triphones	/ **sil-s+p s-p+iy p-iy+ch iy-ch+sil** /

Main problem with Context Dependent phones is trainability.
For N phones

- N^2 biphones
- N^3 triphones etc. (although only a smaller set of these will occur).

Here we are using the HTK notation for context dependency:

- **l-c+r** denotes a phone **c** with a left context of **l** and a right context of **r**

The distributed version of HTK supports the use of word-internal and cross-word triphones. Models of this form (or more complex context dependency) are used in state-of-the-art speech recognition systems.

Context Dependent Phones (2)

Word boundary information may be made use of in defining context.

- **Word-internal**: Word boundaries represent a distinct context.

  "speech task" = / **sil s+p s-p+iy p-iy+ch iy-ch t+ae t-ae+s ae-s+k s-k sil** /

- **Cross-word**. Word boundaries ignored (or used as additional context).

  "speech task" = / **sil-s+p s-p+iy p-iy+ch iy-ch+t ch-t+ae t-ae+s ae-s+k s-k+sil** /

For a 26000 word dictionary, designed for transcription of the Wall Street Journal corpus:

Word Internal	14300 distinct contexts
Cross Word	54400 distinct contexts

Problem: only 22804 cross-word triphones appear in the training data (WSJ SI-284, 66 hours of acoustic data).

System Trainability

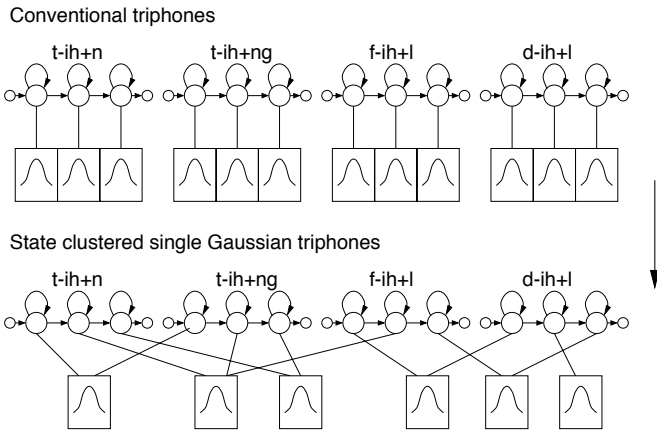
Need to balance between level of acoustic detail and robust parameter estimation.

Backing off

- When insufficient data to train a specific context *back-off* and use a less specific model.
- E.g. *Triphone* $\rightarrow$ *Biphone*$\rightarrow$ *Monophone*
- Inflexible, involves large "jumps" in context dependency

Sharing (Tying)

- *Share* parameters between models that are acoustically similar.
- Flexible as models of the same complexity can share data, as well as with less specific models.
- Can share or **tie** parameters at different levels.
- Figure shows tied output distributions across different triphone contexts of same *base phone*.

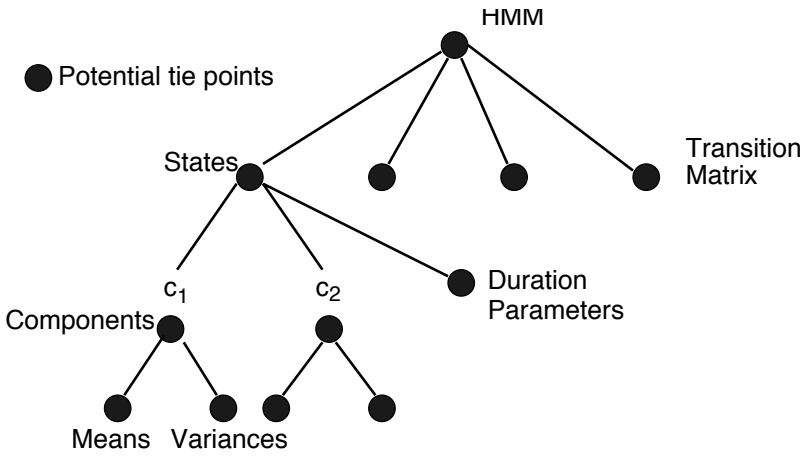


Smoothing & MAP estimation

- Can smooth the parameter estimates for a CD model with a CI model
- **Maximum A-Posteriori** (MAP) estimation (uses the CI models as a prior) gives a well-founded method. MAP estimation be discussed later in the course.
- Can apply in combination with either of the above

Parameter Tying

Parameter tying may be performed at a variety of levels.



For example in HTK:

1. **Generalised Triphones**. Different triphone contexts share the same model.
2. **State-Clustered Triphones**. States of different triphones share same distributions.
3. **Tied Mixture** (or Semi-Continuous HMMs). All output pdfs are shared across all HMMs.
4. **Grand Variance**. The same variance matrix is shared over all Gaussians.

Maximum likelihood training for **Generalised Tying** is implemented by simply pooling the data from all the examples for any particular tied parameter.

Bottom-Up Parameter Tying

For reliable estimates need parameter sharing across contexts. Basic procedure::

1. Models are built for all observed contexts.
2. Merge “models” that are acoustically similar.
3. If sufficient data available **stop** else goto (2).

Two standard forms

- **Generalised Triphones** - The model comparisons and merging may be done at the **model level** to form *Generalised Triphones*.
- **State-Clustered Triphones** - Comparison and clustering may be performed ay the **state level** to form *State-Clustered Triphones*. Allows e.g. left state to be clustered independently of the right state (and centre state).

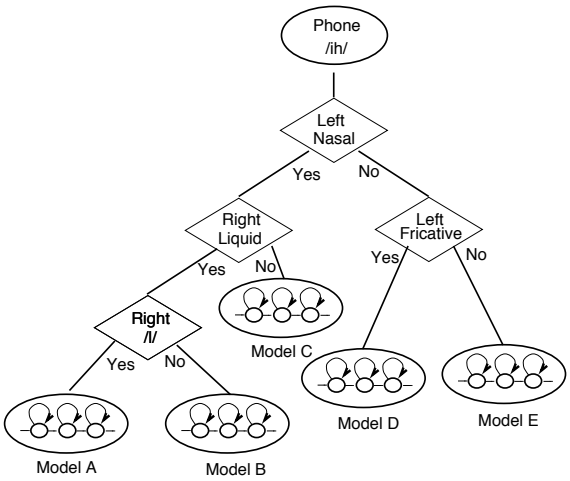
Note that a **distance measure** is needed either between state distributions or between complete models. Can use a measure based on the **Kullback-Leibler** (K-L) number.

Limitations

- Unreliable for contexts that occur rarely in training data.
- Unable (without using back-off) to estimate models for contexts not seen in training data.

Top-Down Parameter Tying

- Binary decision tree
- At each node yes/no decision to form context equivalence classes
- A pre-defined question set of possible is used
- At each node actual question chosen to maximise objective function
- Can be used at model or state level



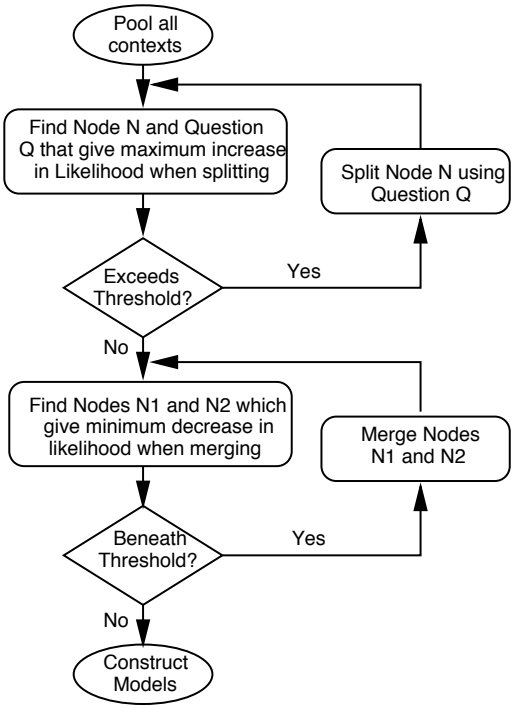
Advantages

1. No need to back-off, handles unseen contexts elegantly.
2. Allows *expert* knowledge to be incorporated (choice of question set)
3. Allows any degree of context dependency to be simply incorporated.

Disadvantages

1. Only locally optimal splits are selected.
2. Not all question combinations normally asked

Constructing Decision Trees



Aim is to grow tree to maximise the increase of (an approximate) likelihood of training data

To make decision tree generation computationally efficient:

1. The frame/state alignment is not altered (hence the contribution of the transition probabilities may be ignored).
2. Single component Gaussians are accurate enough to decide on contexts.

In practice an additional splitting threshold is introduced that ensures there is sufficient data (i.e. state occupation count) at a terminal node for robust estimation of a Gaussian mixture model.

Once tree has been grown a merging process may optionally be used

- Tree leaves are merged if don't decrease objective function (approx. likelihood) too much

Change in Log Likelihood

Change in data log likelihood when splitting into two (single Gaussian) models for particular sets of contexts needed to select best question to ask at each stage of tree construction.

Note that the approximate likelihood will (normally) increase if model the data by two Gaussians rather than one!

If hypothesise a split in the contexts at a particular node, p, into two descendants, r and s, then increase in log likelihood is simply

$$-\frac{1}{2}\log(|\Sigma_s|)N_s - \frac{1}{2}\log(|\Sigma_r|)N_r + \frac{1}{2}\log(|\Sigma_p|)N_p$$

where Σ_p, Σ_r, Σ_s, are covariance matrices associated with the data from node p and splits s and r, and N_p, N_r, and N_s are total number of frames associated with each of the nodes p, s and r so that $N_p = N_r + N_s$.

Note that if mean vector and covariance matrix is stored for each context along with an occupation count, required covariance for any combination of states (i.e. any point in the tree) can be very simply computed.

To compute the change in likelihood for a merge (after splitting is complete) then then a similar likelihood change formula can be used.

Top Down Decision Trees: Summary

- Aim is to group contexts into **equivalence classes**
- Normally have a separate tree for each emitting state position of each phone
- Manually define sensible questions that can be asked
  - Actual questions asked **automatically chosen** to increase approx log likelihood
  - Normally more general questions asked near top of tree (more occurrences)
  - Aim to have questions that will **generalise**
- Use single Gaussian (diagonal) statistics at each stage of tree construction
  - Simple approximation but seems good enough
  - Simplifies computation and stats needed
- Tree splitting is a **greedy** procedure
  - Best split at each step but end result only locally optimal
- Depth of tree will depend on the amount of data available for each state/phone
- When trees generated, tied-state HMMs can be generated for **all contexts**
- Can easily integrate many types of context questions (e.g. longer distance questions, word position dependent) etc.

Building an HMM System

A (fairly simple) large vocab cross-word triphone system may be built as follows:

1. Using best previous models (eg TIMIT models) to obtain phone alignments.
2. Build single Gaussian monophone models (typically 4 re-estimation iterations).
3. “Clone” monophones for every cross-word triphone context seen in training data.
4. Build single Gaussian unclustered triphone models
5. Perform state-level decision-tree clustering: generates initial single-component state-clustered triphones
6. Train single-component models (typically 4 re-estimation iterations)
7. Increase num mix components (iterative component splitting via “mixing-up” procedure)
   - retrain at each level of complexity (typically 4 re-estimation iterations)
   - Number of components increased: 1→2, 2→3, 3→5, 5→7, 7→10, 10→12.

Final system is a **12-component state-clustered cross-word triphone** system.

A Wall Street Journal System (from 1990's)

System details:

1. **Training Data**: Wall Street Journal training data (284 speakers, with 50–150 sentences from each, 36k sentences total, about 66 hours)
2. **Parameterisation**: 12 MFCCs, normalised log-energy, delta and delta-delta parameters.
   - Cepstral Mean Normalisation performed at the per-sentence level.
3. **Acoustic Models**: 12-component GMM-HMM state-clustered cross-word triphones (6.4k distinct states). Maximum likelihood training.
   - Both *Gender-Independent* and *Gender-Dependent* forms.
   - Gender dependent updates mean parameters only gender-dependent subset of training.
4. **Vocabulary**: 65k word vocabulary. Multiple pronunciations.
5. **Language Model**: Trigram language model.
6. **Test Set**: Unlimited vocabulary, "clean" acoustic environment.

System	WER (%)	
	Dev.	Eval
Gender-Independent	9.43	9.94
Gender-Dependent	9.06	9.39

- Continuous speech recognition so errors are either substitutions, deletions or insertions
- Small improvement from male/female model sets.

HTK Toolkit

- HTK (Hidden Markov Model Toolkit): primarily written at Cambridge University Engineering Dept
- Very flexible, modular toolkit written in ANSI C
- Set of source code libraries and tools for building and testing HMM systems
- HTK3: available for download since Sep 2000 `http://htk.eng.cam.ac.uk`
- Have been more than 100,000 users worldwide, heavily used in teaching, and research, active support mailing lists
- Some companies have built models for commercial systems with HTK
- Current public release HTK 3.5 includes major features for GMM-HMM based ASR including a large vocabulary decoder, discriminative training, adaptation
- HTK 3.5 includes fully integrated support for artificial neural network (ANN) models
- Documentation uses the HTK Book
- New release (HTK 3.5.1) in preparation which includes significantly expanded range of ANN models that can be used
  - Practicals use a pre-release version of HTK 3.5.1 and PyHTK library
  - HTK book for HTK 3.5 avalaible
  - **Version available for practicals is for use within CUED only!**
- Other extensions - e.g. HTS for HMM-based synthesis

Speech Recognition Practical: TIMIT Speech Recognition

- Phone model training/testing using TIMIT acoustic-phonetic database
- Uses HTK tools for training/testing models
- Initial part of practical focuses on GMM-HMM models
- Set of scripts (written in either C-shell or Python) and resources provided so that you can easily train/test
  - phone models with different initialisations (TIMIT timings vs flat-start)
  - different speech parameterisations (log mel filterbank, MFCCs)
  - effect of adding differential coefficients to the front-end
  - Gaussian mixture models of varying complexity
  - Context-independent and triphone models via decision tree tying
- Later parts of practical deal with feed-forward DNN-HMMs and recurrent models (RNN-HMMs).
- In addition there are possible extensions (depending on time etc) including:
  - use of a bigram language model rather than a simple phone loop
  - extended forms of recurrent models
- The writeup requires a description of experiments and discussion/analysis of results.
- Demonstrated practical sessions (initially on-line)
- Write-up due before Lent term lectures start (date on handout)
- Practical handout on MLMI2 moodle page

Summary

- Phone-based sub-word units are used as acoustic modelling units
- Context-dependent units are needed to reduce variability due to co-articulation
- Control parameter numbers by backing-off or by parameter sharing
- Standard approach is now to use decision-tree based top-down state tying
  - Uses human language knowledge defining question sets
  - Automatically chosen questions when building trees
  - Efficient due to simple single Gaussian assumptions
  - Context equivalence classes with enough data for robust estimation
- WSJ Speaker independent large vocabulary speech recognition using these principles
  - Trained on about 60hours of read speech in low noise environment
  - Gives word error rates of less than 10%
  - Lower if more acoustic training data, adaptation, more advanced training techniques, better language models, DNN acoustic models etc.
- Speech recognition practical on TIMIT phone recognition:
  - Experiments to investigate various aspects of performance of HMM speech recognition design
  - Initial parts of experiment use GMM-HMMs, later parts use DNN-HMMs and RNN-HMMs