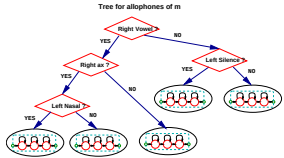
COM4511/6511 Lecture 8
Acoustic modelling with HMMs



Thomas Hain

19th February 2020

SPANDH

Outline for this lecture

- Acoustic units
  - Properties
  - Whole-word models vs phone models
- Generalisation
- Context dependent models
  - Triphones
- Parameter tying
  - Bottom-up
  - Top-Down

Thomas Hain

From a classification point of view units should have the following properties:

- **Compact**
  Ideally the number of parameters in the set of models should be independent of the vocabulary size, task complexity etc.
- **Representative**
  The model for a unit should in theory be able to capture unit specific variation only, units need to be defined accordingly.
- **Clear relationship to words**
  The relationship to words should be well defined, i.e. clear mappings that allow for example the construction of models for unseen words.
- **Distinctive**
  Units should be well distinguishable from each other (discriminative!)
- **Clear topology**
  The model topology (#states, transitions) should be easily definable. Ideally each unit has the same topology.

Comparison Word models - Phone models

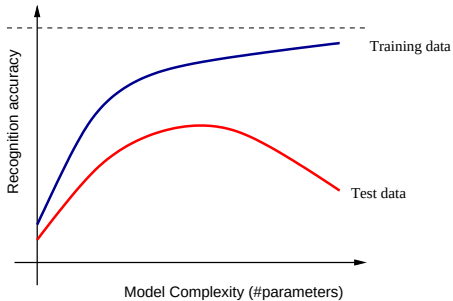
Using the above criteria we can compare word and (mono-)phone models:

	Word models	Phone Models
Compact	#models increases linearly with #words	#models is constant
Representative	Implicitly capturing all variation in terms of pronunciation	Do not capture variation in pronunciation and coarticulation
Word mapping	trivial new words need to be trained	via dictionary
Distinctive	These are exactly the classes we want to separate	The monophones are broad classes that overlap highly in acoustic space
Topology	To be determined heuristically	Standard fixed #states is sufficient even though some tuning may help
#Parameters	linear with the #words	a function of the number of phones

The number of free model parameters necessary for representing the acoustic models is of great importance.

Generalisation

Unit selection is driven by the requirement that the units should be trainable, i.e. that sufficient data for each unit is available. Implicitly, the number of units defines the number of parameters present in a model.



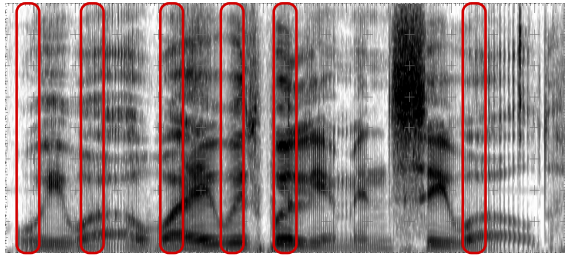
The number of parameters needs to be optimised to yield optimal performance. As can be seen from the figure to the left, the performance on the training set will improve steadily with an increase in the number of parameters. In contrast the performance on the test set decreases after a certain model complexity is reached. This effect is called **over-training**.

The optimal number of models and number of free parameters are training set dependent!

Thomas Hain

Coarticulation

Coarticulation means that the actual sound associated with a particular phoneme depends on the neighbouring sounds (in a word or sentence context).



The diagram shows a spectrogram for the phrase:

We were away with William in Sea World

Evidently the spectrograms differ substantially of each realisation of *w*. To handle this problem **context dependent phone models** have to be used.

Models for phones are **context independent** and are called **monophones**. Models that take the phonemic neighbours into account are called **context dependent**. The various models associated with one particular phoneme are called **allophones**. Several types of models with varying context depth exist:

Unit names	Units
Word	**SPEECH**
Monophones	**s p iy ch**
Left Biphones	**sil-s s-p p-iy iy-ch**
Right Biphones	**s+p p+iy iy+ch ch+sil**
Triphones	**sil-s+p s-p+iy p-iy+ch iy-ch+sil**

This can be extended to ± 2 neighbours (pentaphones). Note that the dictionary still contains the monophone representation. The conversion into triphone names is performed automatically. The most commonly used models are **triphones**.

Triphones: Practical issues

The number of potential phone models grows exponentially with the context depth.

Units	Factor	#Models
Word	M	M
Monophones	N	46
Left Biphones	N^2	2116
Right Biphones	N^2	2116
Triphones	N^3	97336
Pentaphones	N^5	205962976

Number of potential phone models

The large number of potential models causes two problems, both due to limited amounts of training data:

P1 **Insufficient data** for training of an individual triphone model is observed in the training data.

P2 The training and test dictionaries differ normally. **Unseen triphones** are those that would be required for recognition, but are not present in the training data.

One can limit the number of models required by using **word-internal triphones**:

SPEECH TASK **s+p s-p+iy p-iy+ch iy-ch t+ae t-ae+s**

instead of **crossword triphones**:

Word-internal triphones make use of lower order models and thus have the advantage of much smaller amounts of distinct contexts. For example in the case of WSJ, a read speech corpus, the number of distinct crossword triphones in the training data is 54400 whereas for word-internal triphones it is only 14300. However, it results in **poor modelling of very short words**, which are normally frequent. Further in continuous speech normally there are no pauses and hence **coarticulation** occurs across word boundaries. The **added cost for search** however is **low**.

We have to find strategies to deal with problems P1 and P2 !

The simplest strategy to deal with problem **P2** is to back off to models that have been observed in the training data and thus could be trained.

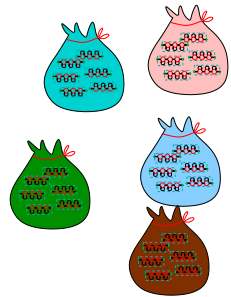
1. Search for triphone X-Y+Z in the model set.
2. If not avaialble search for Y+Z or X-Y.
3. If both are not available use the monophone model Y.

This strategy is called model **back-off**. For discrete HMMs sometimes interpolation is used.

Problem **P1** can only be controlled by defining units for which sufficient data is available. This can be done by grouping of several triphones together to form clusters.

Two strategies are commonly used:

1. **Clustering by rule**: Groups of triphones are formed on the basis of manually generated rules.
2. **Data driven clustering**: The clusters are formed automatically by optimisation of some distance criterion.



Data driven clustering is by far the better choice as it allows automatic adjustment of the model set sizes to the properties of the training data (small number of clusters for small amounts of data, large numbers for large amounts of data). Note that clusters do not have to be of equal size. We distinguish between bottom-up and top-down clustering.

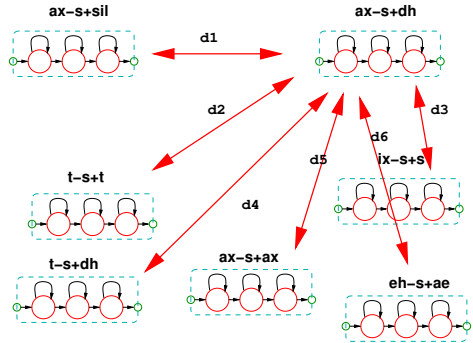
Bottom-up Agglomerative Clustering

Bottom-up clustering requires that first all triphone models are trained in the standard fashion.

Then the following steps are taken:

1. Compute the distance of all triphones to each other.
2. Merge the tow triphones that are closest.
3. Repeat the procedure until the distance exceeds a certain threshold.

Merging two models produces another model that is now representative of both contexts.



Note that the tying (clustering) of models is not necessarily restricted to complete triphone HMMs (generalised triphones). It turns out that the **clustering** is much more powerful (i.e. yields far better results) if performed **on the state level**. This means that only the output probability distributions (GMMs) are clustered.

Clustering requires the definition of a distance metric, i.e. a measure how different a pair of triphone models really is. In practice the definition of such a distance metric is difficult. Typically one uses **likelihood based distance measures** such as

$$d(\mathcal{M}_1, \mathcal{M}_2) = \frac{1}{T^1} \left(\log \left(p(\mathbf{O^1}|\mathcal{M}_1) - \log \left(p(\mathbf{O^1}|\mathcal{M}_2) \right) \right) \right)$$

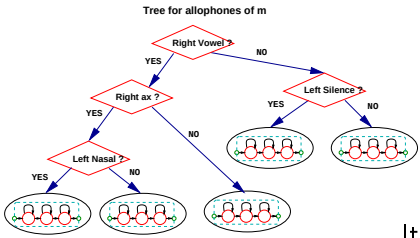
where $\mathcal{M}_{1,2}$denote models and $\mathbf{O}^1$ is a sequence of T^1vectors generated by model $\mathcal{M}_1$. For the simplified case of 2 single Gaussian models this turns into the so-called Kullback-Leibler distance:

$$d(\mathcal{M}_1, \mathcal{M}_2) = \frac{1}{2} \left(\mathrm{tr}\left(\boldsymbol{\Sigma}_2^{-1} \boldsymbol{\Sigma}_1 - \mathbf{I} \right) \right) + \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \right) \boldsymbol{\Sigma}_2^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \right) + \log \left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|} \right)$$

Note that the above is not symmetric, a symmetric version is given by

$$d'(\mathcal{M}_1, \mathcal{M}_2) = \frac{d(\mathcal{M}_1, \mathcal{M}_2) + d(\mathcal{M}_2, \mathcal{M}_1)}{2}$$

A **binary decision tree** based on phonetic questions about neighbouring phonemes can be used for Top-Down clustering. In contrast to agglomerative clustering this allows not only to solve problem **P1**, but also **P2** (unseen triphones).



- No need for backoff, unseen contexts handled seamlessly
- Expert knowledge in the form of questions
- Allows to incorporate (at least in theory) arbitrary contexts
- Scales to the amount of training data
- Usable for HMMs and individual states

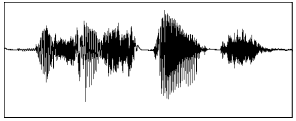
It is standard to use phonetic decision trees on a state-level in conjunction with a likelihood criterion formulated on the basis of the training data (different to before).

At each stage in the top-down clustering procedure each question out of a predefined (manually generated) set is tested (each question can split a set of triphones in two parts). If a sufficient gain in likelihood is obtained by the split the question is added to the tree and the process repeated.

Training Sub-word Models

The training of sub-word units is a straightforward extension from the training of continuous speech whole-word HMMs.

Each training set consists of pairs of audio signals together with word level transcripts:



This is speech

Before standard Baum-Welch training can resume the appropriate sentence models need to be constructed (see right)

- **Whole-word models**: Each word is replaced by the appropriate left-to-right model for each word. All models are concatenated.

- **Monophone models**: Each word is first replaced by the string of phones and then by the models for each phoneme.

- **State-clustered triphone models**: Each word is first replaced by a string of phones, then the phones are converted into triphones. Finally for each triphone state the appropriate output distribution is found by use of a phonetic decision tree.

Thomas Hain