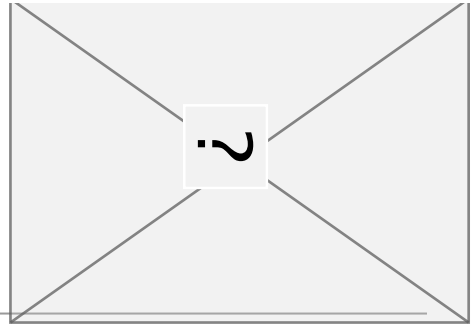
HMM-based speech synthesis

Simon King
Centre for Speech Technology Research
University of Edinburgh, UK

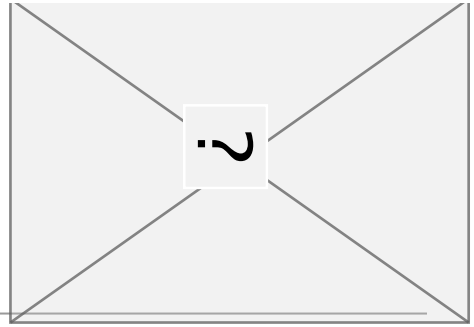
www.cstr.ed.ac.uk

Acknowledgements



- The work and ideas presented here benefited from the help and advice of many people, including
  - Keiichi Tokuda and colleagues at the Nagoya Institute of Technology (NIT)
  - colleagues at the Centre for Speech Technology Research in Edinburgh
  - partners in the EMIME project

Outline

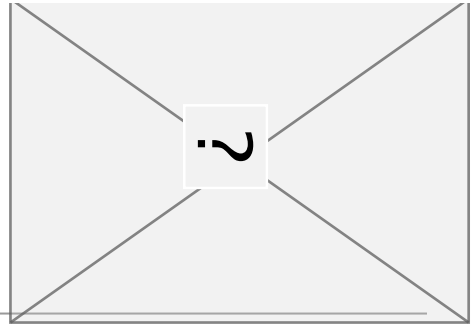


- HMM-based speech synthesis
- Brief detour - the units of speech, context-dependency, clustering
- Unsupervised adaptation for synthesis: experiments, examples, results

- Talks later today that continue this theme:
  - John Dines - the gap between HMM-based recognition and synthesis
  - Junichi Yamagishi - latest results on speaker-adaptive synthesis

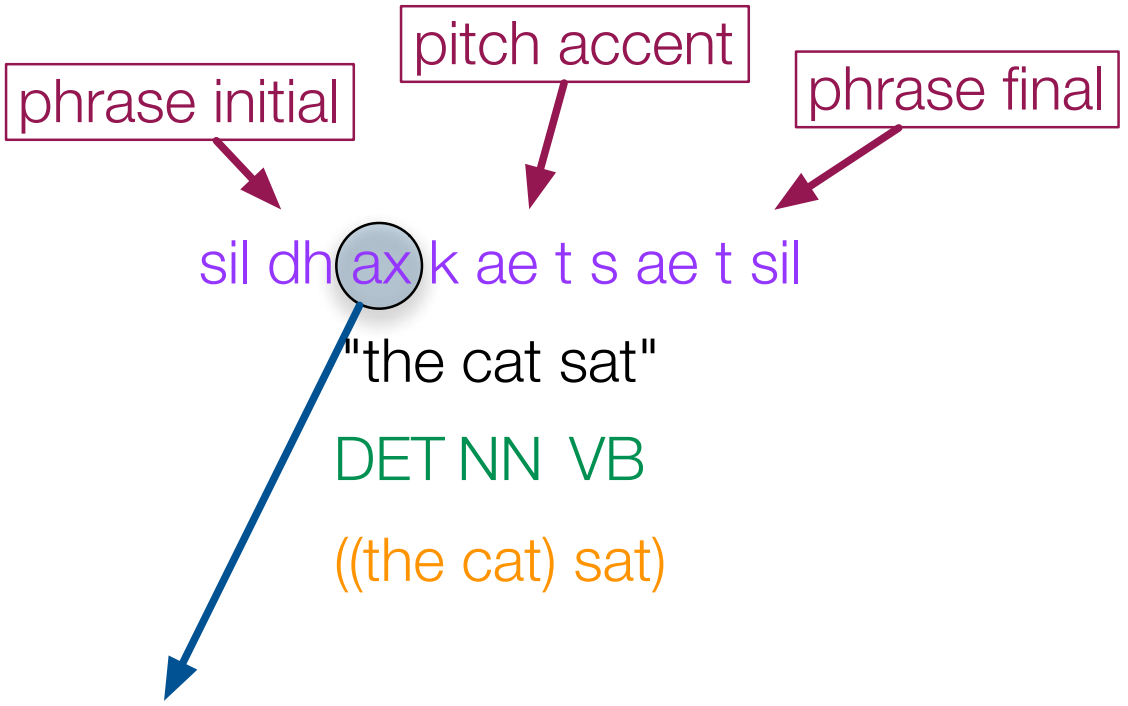
HMM-based speech synthesis in a nutshell

Speech synthesis mini-tutorial

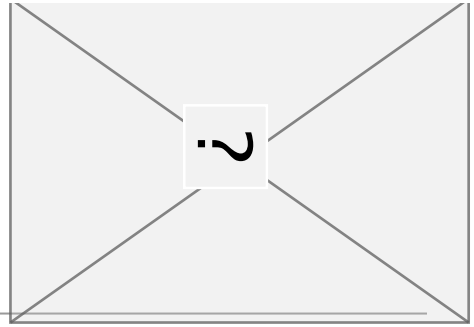


- Text to speech
  - *input:* text
  - *output:* a waveform that can be listened to
- Two main components
  - *front end:* analyses text and converts to linguistic specification
  - *waveform generation:* converts linguistic specification to speech

From words to linguistic specification

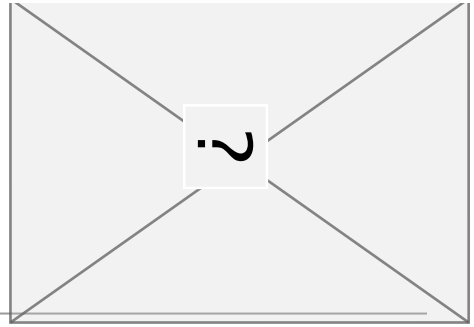


From linguistic specification to speech



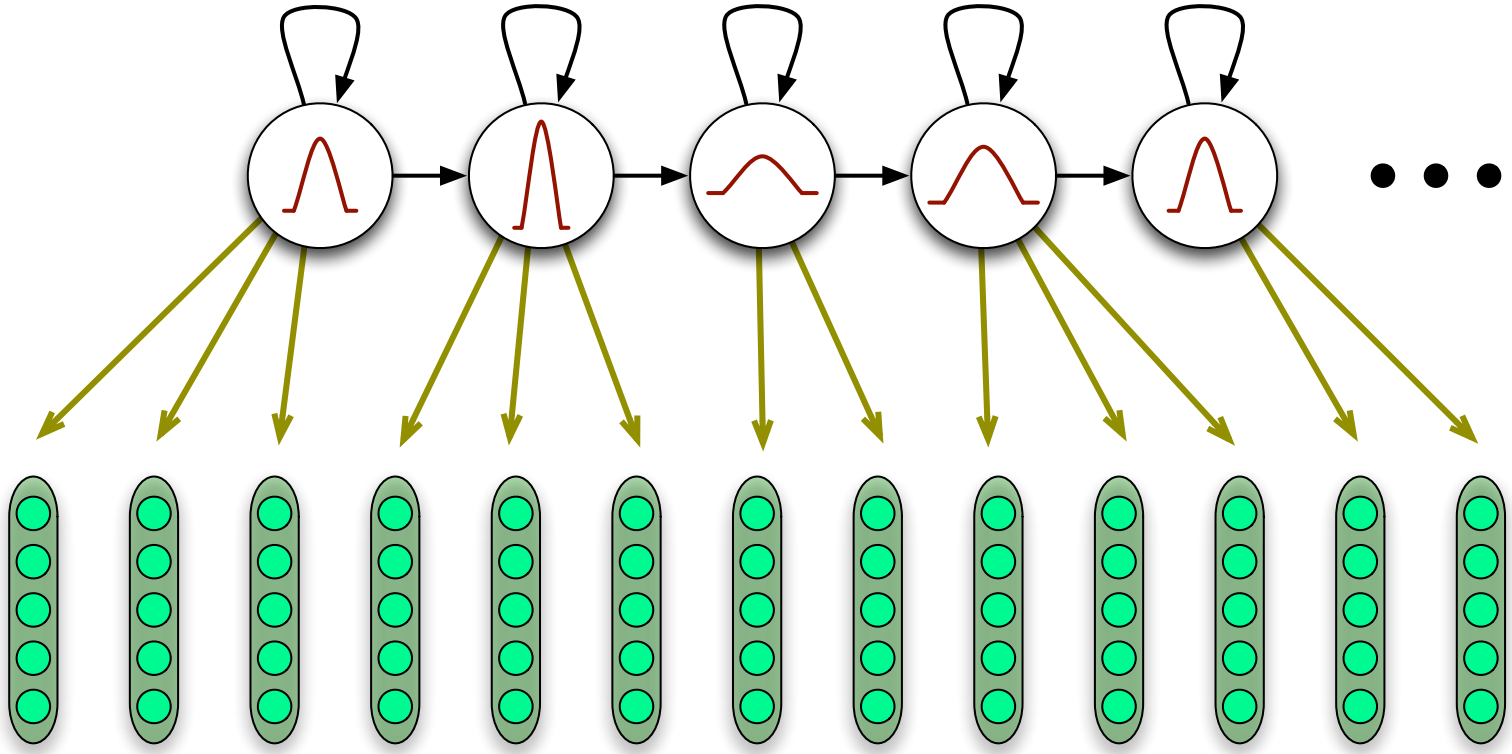
- Two possible methods
  - Concatenate small pieces of pre-recorded speech
  - Generate speech from a model

HMM-based speech synthesis mini-tutorial

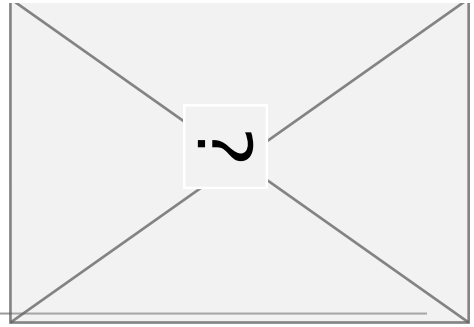


- HMMs are used to generate sequences of speech (in a **parameterised form** that we call ‘speech features’)
- From the **parameterised form**, we can generate a waveform
- The **parameterised form** contains sufficient information to generate speech:
  - spectral envelope
  - fundamental frequency (F0) - sometimes called ‘pitch’
  - aperiodic (noise-like) components (e.g. for sounds like ‘sh’ and ‘f’)

HMMs are generative models

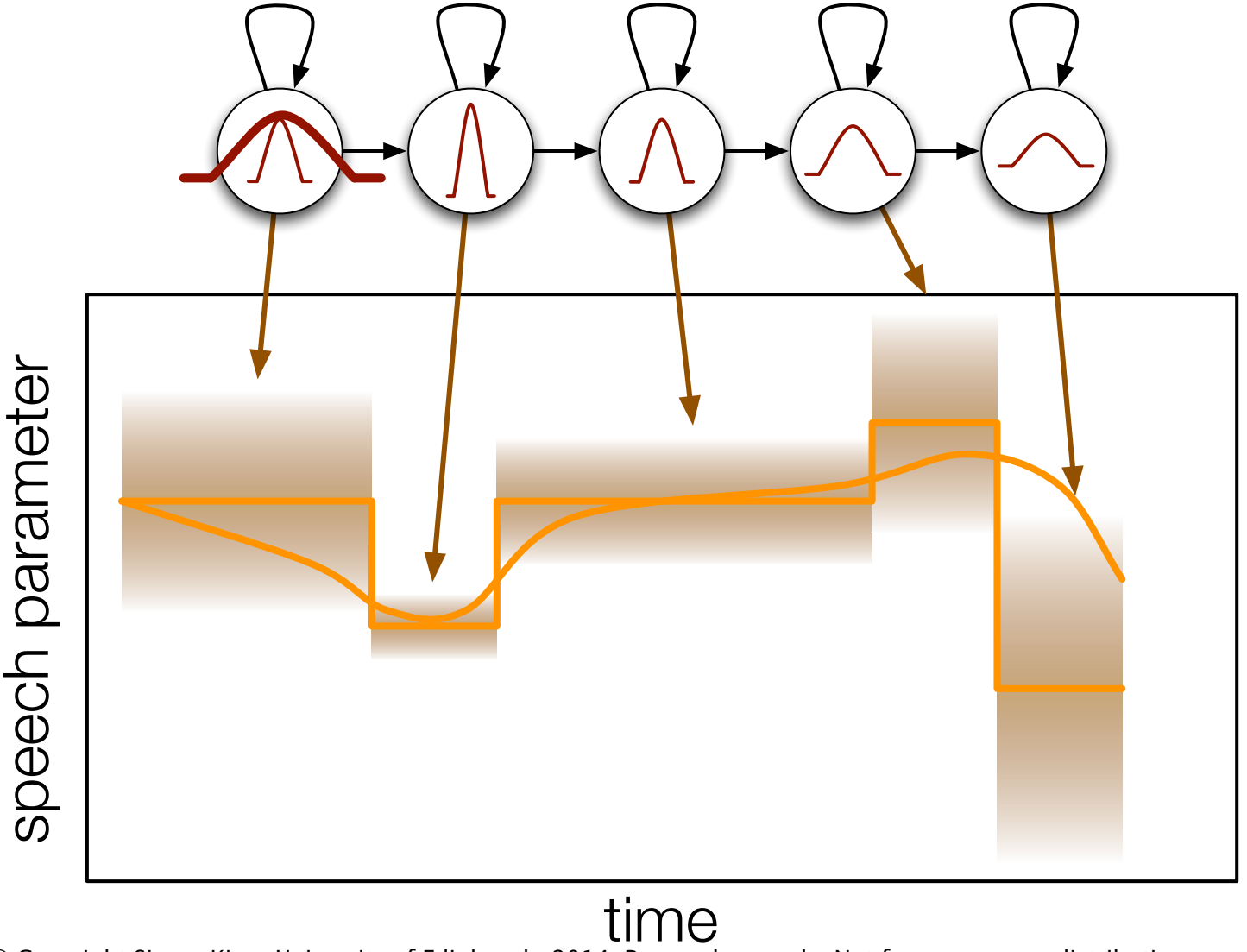


Trajectory HMMs

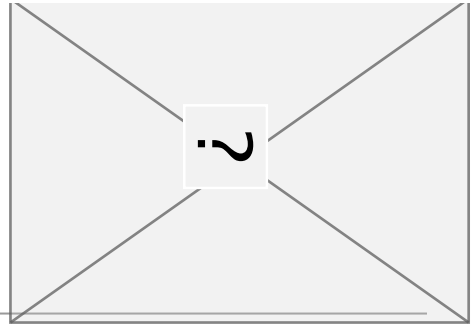


- Using an HMM to generate speech parameters
  - because of the Markov assumption, the most likely output is the sequence of the means of the Gaussians in the states visited
  - this is piecewise constant, and ignores important dynamic properties of speech
- Trajectory HMM algorithm (Tokuda and colleagues)
  - solves this problem, by correctly using statistics of the dynamic properties during the generation process

Trajectory HMMs



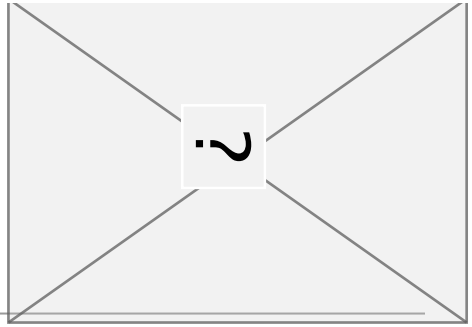
Constructing the HMM



- Linguistic specification (from the front end) is a sequence of phonemes, annotated with contextual information
- There is one 5-state HMM for each phoneme, in **every required context**
- To synthesise a given sentence,
  - use front end to predict the linguistic sp on
  - concatenate the corresponding H
  - generate from the HMM

Sparsity problem!

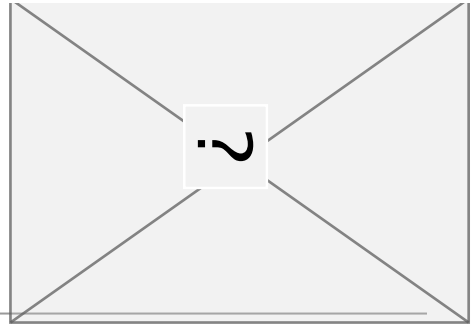
Example linguistic specification




```
pau^pau-pau+ao=th@x_x/A:0_0_0/B:x-x-x@x-x&x-x#x-x$.....
pau^pau-ao+th=er@1_2/A:0_0_0/B:1-1-2@1-2&1-7#1-4$.....
pau^ao-th+er=ah@2_1/A:0_0_0/B:1-1-2@1-2&1-7#1-4$.....
ao^th-er+ah=v@1_1/A:1_1_2/B:0-0-1@2-1&2-6#1-4$.....
th^er-ah+v=dh@1_2/A:0_0_1/B:1-0-2@1-1&3-5#1-3$.....
er^ah-v+dh=ax@2_1/A:0_0_1/B:1-0-2@1-1&3-5#1-3$.....
ah^v-dh+ax=d@1_2/A:1_0_2/B:0-0-2@1-1&4-4#2-3$.....
v^dh-ax+d=ey@2_1/A:1_0_2/B:0-0-2@1-1&4-4#2-3$.....
```

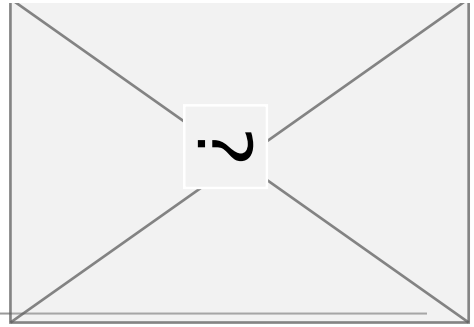
"Author of the ..."

HMM-based speech synthesis



- Differences from automatic speech recognition include
  - Synthesis uses a much richer model set, with a lot more context
    - For speech recognition: triphone models
    - For speech synthesis: “full context” models
  - “Full context” = both phonetic and prosodic factors
  - Observation vector for HMMs contains the necessary parameters to generate speech, such as spectral envelope + F0 + multi-band noise amplitudes

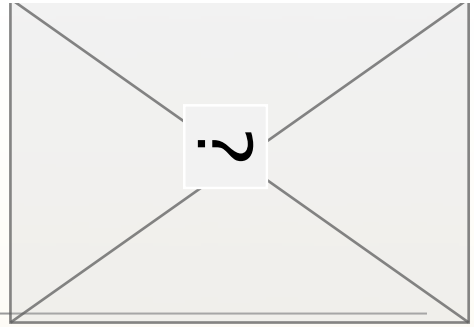
Comparison with ASR



- Differences from automatic speech recognition include
  - Synthesis uses a much richer model set, with a lot more context
    - For speech recognition: triphone models
    - For speech synthesis: “full context” models
  - “Full context” = both phonetic and prosodic factors
  - Observation vector for HMMs contains the necessary parameters to generate speech, such as spectral envelope + F0 + multi-band noise amplitudes

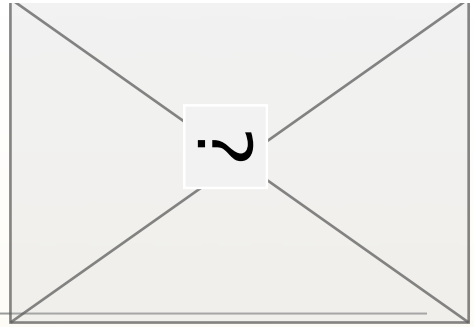
Brief detour - speech units

Interchangeability, equivalence, re-use



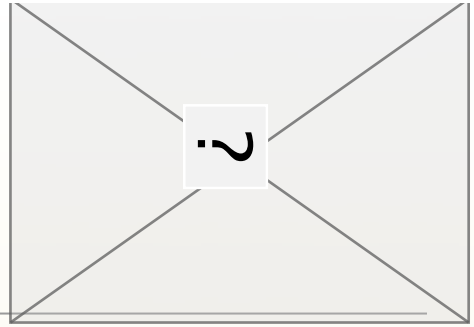
- For speech synthesis, we are making new utterances out of old
  - We need to find interchangeable (re-usable) units of speech
- For speech recognition, we want to recognise previously-unseen words
  - We need to find units of speech that give similar-sounding words similar pronunciations: equivalence (re-use) of classes
- The problems are the same, so might the method for solving them be the same?

Speech structure



- Multi-level / tiered / hierarchical ....
- Broad agreement about the general levels: phonetic, phonological, prosodic
- Even agreement about some units: phones, phonemes, syllables, prosodic phrases, ...
- But no definitive model of how they interact
- The conventional solution is to
  - flatten the structure
  - then use data to find the interactions

Flattening the structure



Phrase Phrase

Word Word Word

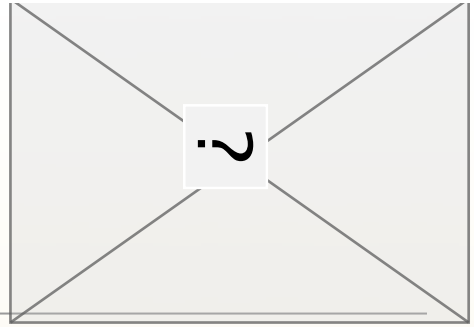
Pitch accent

Boundary tone

Syllable Syllable Syllable Syllable

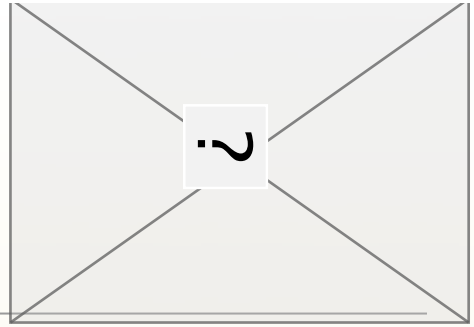
P P P P P P P P P P P P P P P P

Context



- Flattened representation results in context-dependent phonemes
- The context is very rich, especially for speech synthesis
  - What ***factors*** of the context are important?
  - How many different ***values*** can each factor take?
  - What effect do factors have ***in combination?***
- Some contexts will be *indistinguishable* in terms of the effect they have acoustically (or perceptually)
  - How many *different* contexts are there?
  - Use data and clustering based on context features (typically a decision tree where each leaf is a cluster of models) to learn this
- This is a powerful technique, but has limitations

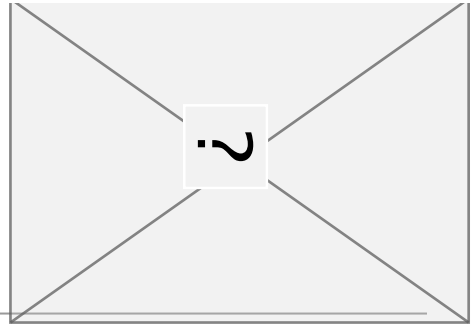
Decision trees have limitations



- Clustering parameters works: good results in both recognition and synthesis
- But, underlying model implied by the clustered parameters is limited
- In particular, ‘divide and conquer’ tree-based clustering:
  - does not handle factorial effects well (e.g., [s], [z], [f], [v] could be modelled with 2 ‘additive’ factors - noise spectral shape & voicing instead of 4 decision tree leaves)
  - can produce leaves which are distinct in terms of context features, but are acoustically similar - a poor use of the data
- Of course we, can construct questions to reflect prior knowledge of speech structure (e.g. groupings of high vowels, voiced fricatives,...) but this is not a complete solution

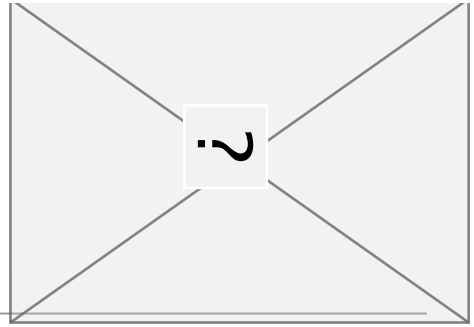
Speaker adaptation

Model adaptation



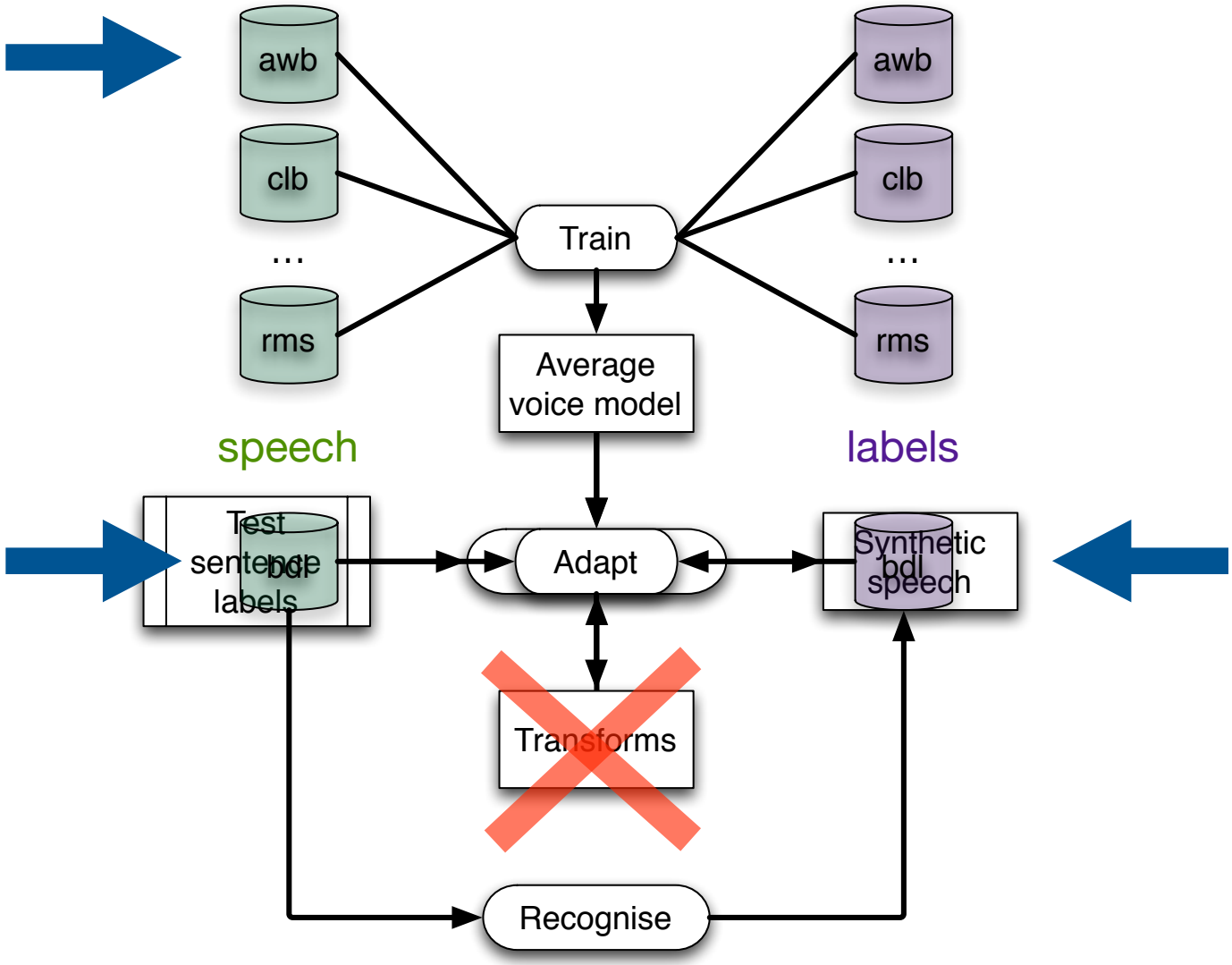
- Training the models needs 1000+ sentences of data from one speaker
- What if we have insufficient data for this target speaker?
- Adaptation:
  - Train the model on lots of data from other speakers
  - Adapt the trained model's parameters using a small amount of target speaker data
    - estimate linear transforms to maximise the likelihood (MLLR)
    - also in combination with MAP

Speaker adaptation using linear regression

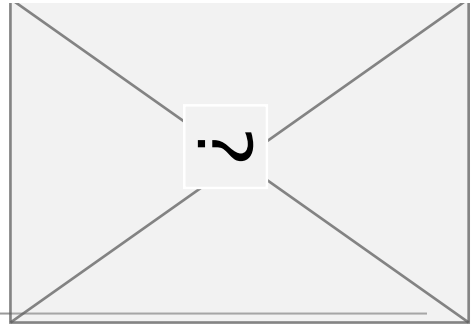


- One of the most important recent developments in speech recognition
- A linear transform (i.e. multiplication by a matrix) is applied to every HMM parameter (Gaussian mean and variance) in order to adapt the model to new data
- Can be used to create new voices for speech synthesis:
  - Train HMM on lots of data from multiple speakers
  - Adapt this HMM using a small amount of data from the target speaker
- This is a very exciting development in speech synthesis

Training, adaptation, synthesis

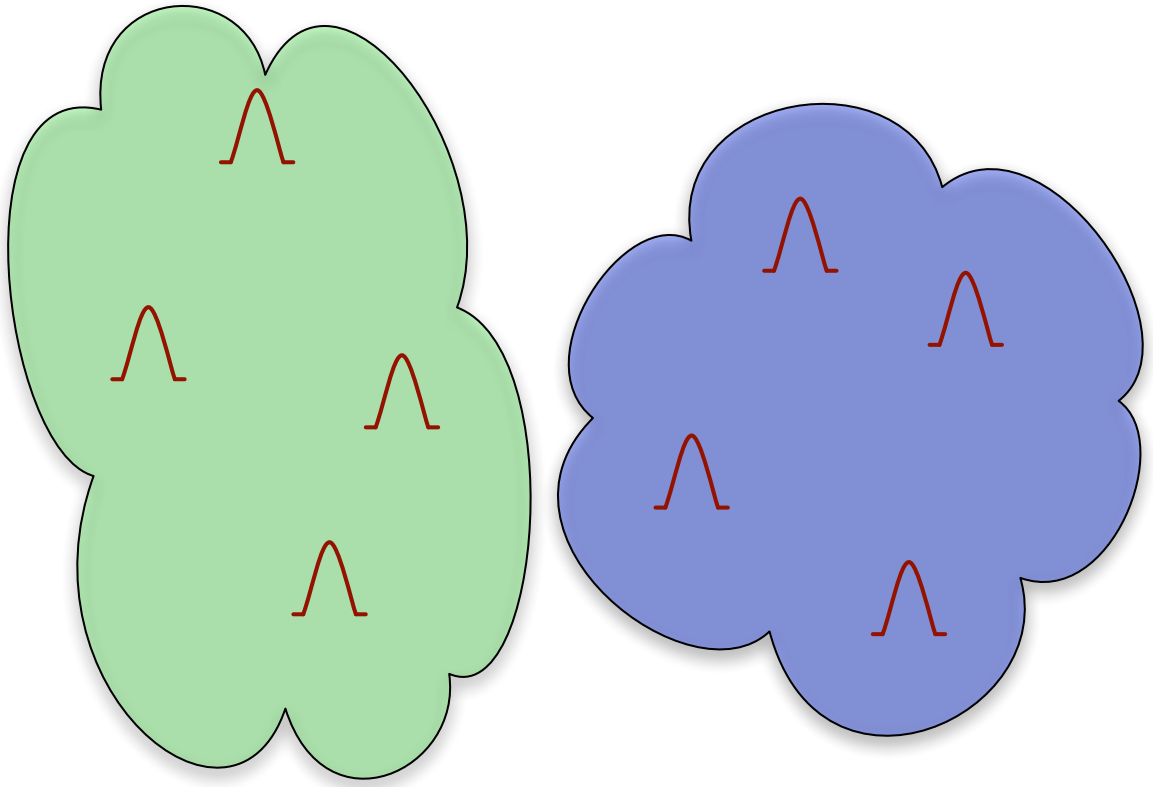


Regression classes

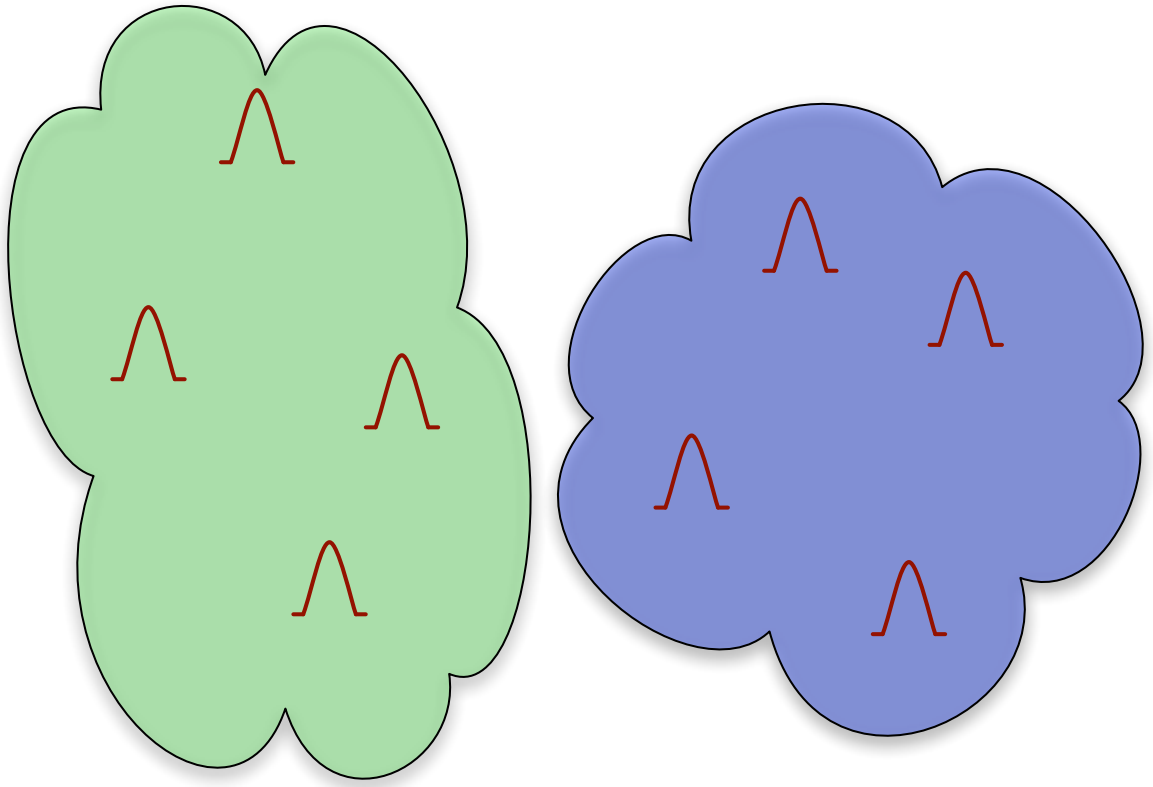


- Amount of adaptation data is limited
- Therefore, will not see examples of all models
- How can we transform the parameters of unseen models?
  - share transformation matrices amongst groups of parameters
- These groups are called “regression classes”
- How many regression classes should we use?
  - it depends on the amount of data available

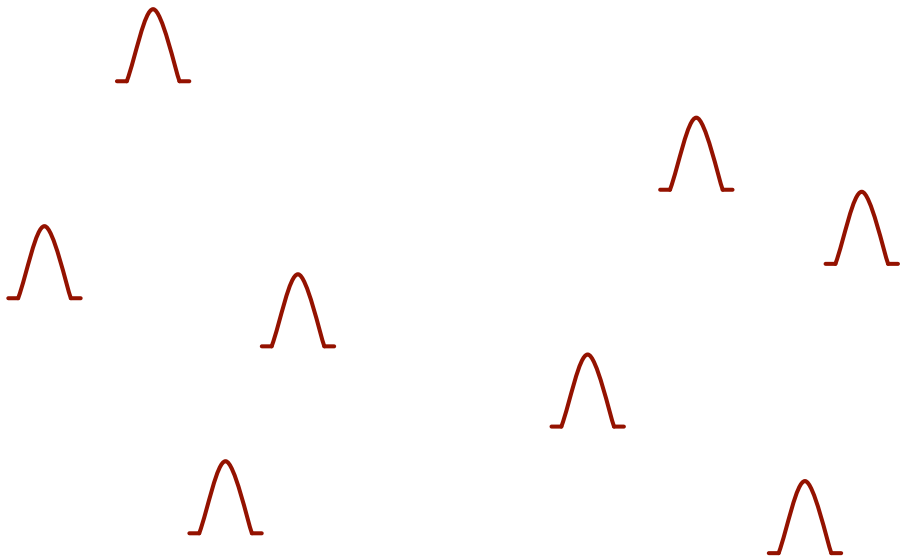
Linear transforms are applied to classes



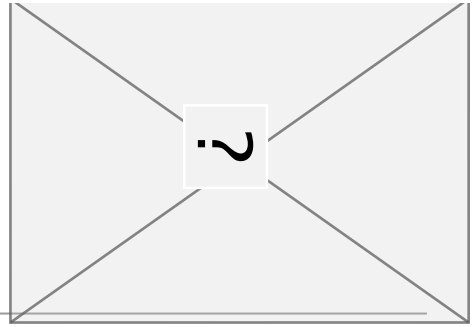
Linear transforms are thus applied to parameters



The effect is a transform of the whole space

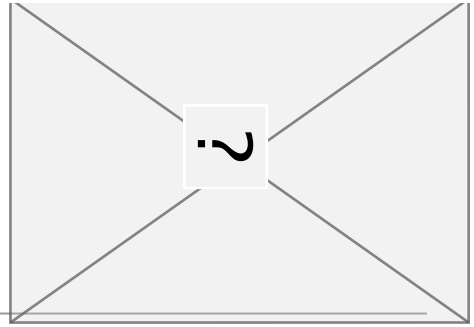


Adaptation transforms the whole acoustic space



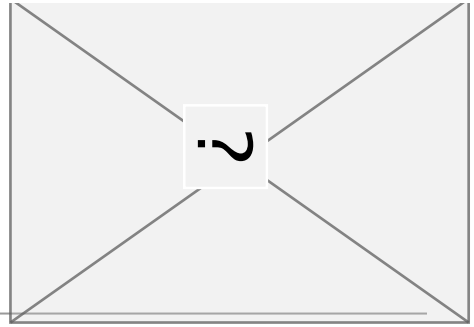
- Ideally, we want to learn a complex, non-linear transformation of the whole acoustic space, in which the model parameters live
- Class-based adaptation is a locally-linear approximation to this
- Using more classes allows more complex adaptation of the whole space
- Key points:
  - because the whole space is transformed, ***all*** parameters are transformed
  - ***any other model*** that lives in the same space can also be transformed

Unsupervised speaker adaptation



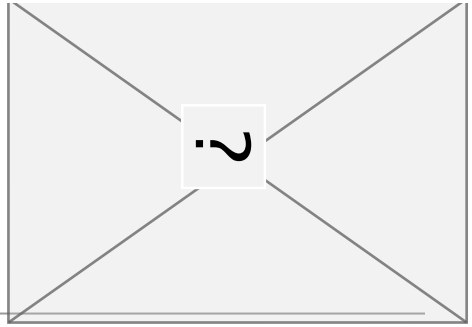
- Same as supervised adaptation
- But the labels are obtained automatically using speech recognition
- Recognition will have errors, but adaptation is still effective

The main issue for **unsupervised** adaptation of speech synthesis models



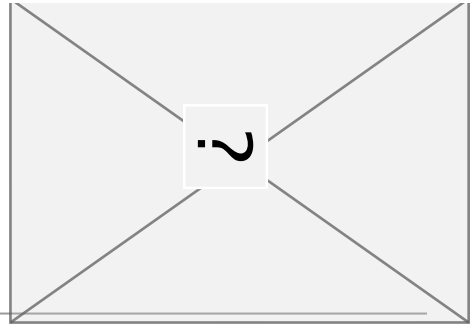
- To directly adapt synthesis models, we need ***full context labels for the adaptation data***
  - Can we get these labels automatically?
  - For now, we presume this is not possible
- However, we can get phonetic labels automatically
  - Can we adapt full context models using phonetic models / labels?

Full context labels



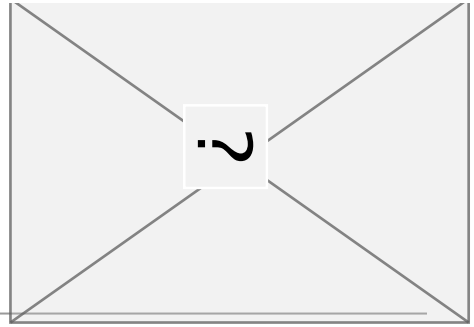

```
pau^pau-pau+ao=th@x_x/A:0_0_0/B:x-x-x@x-x&x-x#x-x$.....
pau^pau-ao+th=er@1_2/A:0_0_0/B:1-1-2@1-2&1-7#1-4$.....
pau^ao-th+er=ah@2_1/A:0_0_0/B:1-1-2@1-2&1-7#1-4$.....
ao^th-er+ah=v@1_1/A:1_1_2/B:0-0-1@2-1&2-6#1-4$.....
th^er-ah+v=dh@1_2/A:0_0_1/B:1-0-2@1-1&3-5#1-3$.....
er^ah-v+dh=ax@2_1/A:0_0_1/B:1-0-2@1-1&3-5#1-3$.....
ah^v-dh+ax=d@1_2/A:1_0_2/B:0-0-2@1-1&4-4#2-3$.....
v^dh-ax+d=ey@2_1/A:1_0_2/B:0-0-2@1-1&4-4#2-3$.....
```

Could we do it *without* full context labels?



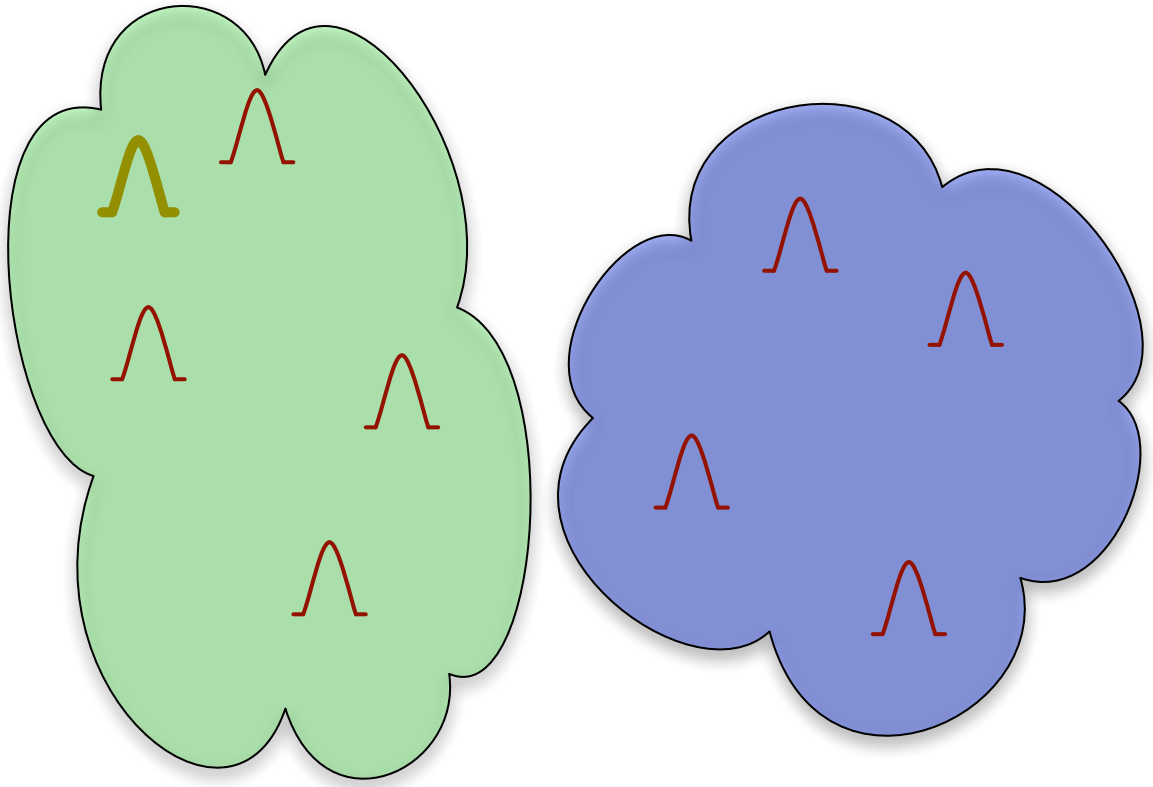
- If the full context model lives in the same acoustic space as the triphone model, then it could use the same adaptation transforms
- Formulate the problem in terms of adaptation transforms
- Adaptation transforms consist of:
  - Matrices used to ***transform*** means (and possibly variances) of Gaussians
  - A grouping of model parameters into regression ***classes***. Each class has one transform

Using triphone models to adapt full context ones

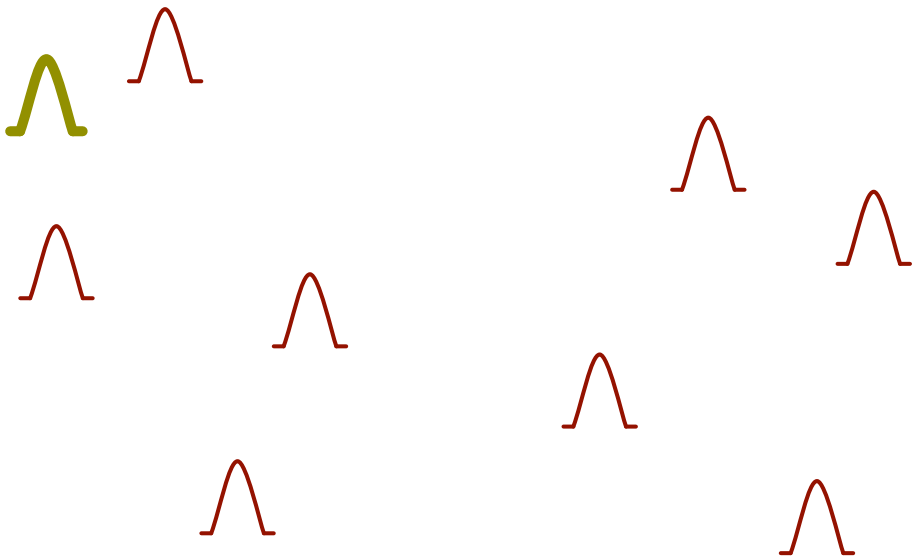


- The problem can be restated as:
  - How can recognition (i.e. triphone) and synthesis (i.e. full context) models use the ***same adaptation transforms***?
- Answer:
  - there must be a single set of regression classes
  - every triphone model parameter belongs to exactly one regression class
  - every full context model parameter belongs to exactly one regression class

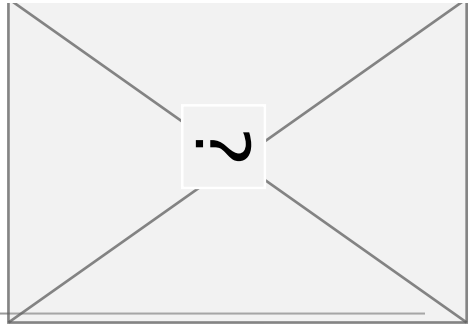
Transforming parameters from other models



Transforming parameters from other models

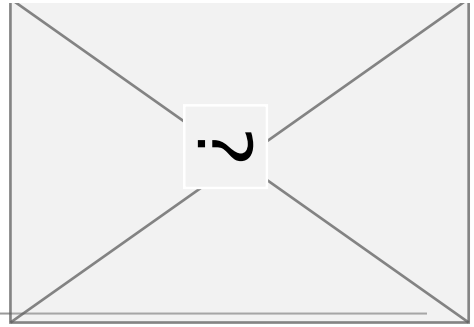


Using triphone models to adapt full context models



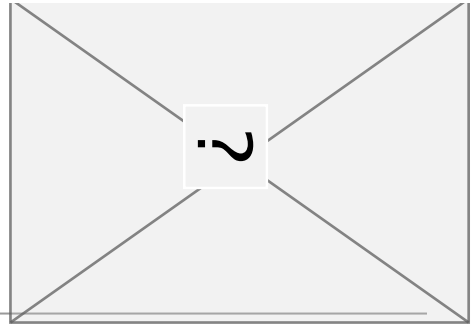
- Triphone models will be used to learn the adaptation matrices
- These matrices will then be applied to full context models
- Which regression class will a full context model parameter belong to?
  - The same class as the corresponding triphone model parameter
- Assumes that the acoustic space transformation learned for triphones is appropriate for full context models - this seems reasonable
- Since the ***classes*** were learned for triphone models, prosodic ***distinctions*** cannot be made. However, global effects (e.g. pitch range transformation) can be achieved

Full context labels would be hard to obtain automatically:



```
pau^pau-pau+ao=th@x_x/A:0_0_0/B:x-x-x@x-x&x-x#x-x$.....
pau^pau-ao+th=er@1_2/A:0_0_0/B:1-1-2@1-2&1-7#1-4$.....
pau^ao-th+er=ah@2_1/A:0_0_0/B:1-1-2@1-2&1-7#1-4$.....
ao^th-er+ah=v@1_1/A:1_1_2/B:0-0-1@2-1&2-6#1-4$.....
th^er-ah+v=dh@1_2/A:0_0_1/B:1-0-2@1-1&3-5#1-3$.....
er^ah-v+dh=ax@2_1/A:0_0_1/B:1-0-2@1-1&3-5#1-3$.....
ah^v-dh+ax=d@1_2/A:1_0_2/B:0-0-2@1-1&4-4#2-3$.....
v^dh-ax+d=ey@2_1/A:1_0_2/B:0-0-2@1-1&4-4#2-3$.....
```

But, by using triphone models, we only need phonetic labels:



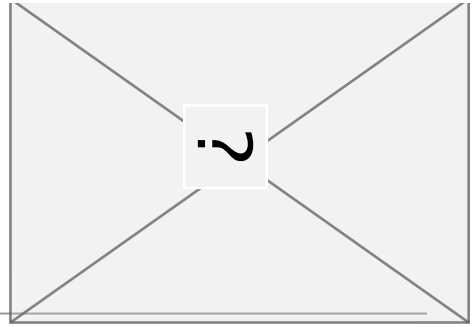
```
pau
ao
th
er
ah
v
dh
ax
```

...which can be obtained automatically.

Unsupervised adaptation is now possible!

Experiments

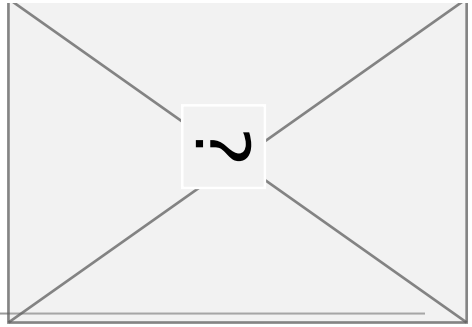
Experiments



- Full results are reported in:

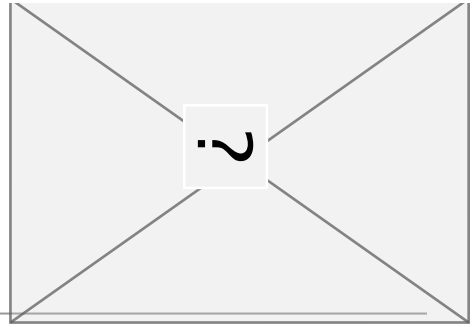
Simon King, Keiichi Tokuda, Heiga Zen, Junichi Yamagishi, "Unsupervised adaptation for HMM-based speech synthesis" in Proc. Interspeech 2008, Brisbane, Australia.

Data



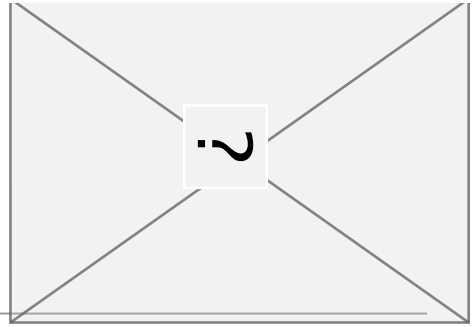
- ARCTIC corpus
  - each speaker reads about 1130 phonetically balanced sentences
- Training speakers: 5 speakers (3 male, 2 female / 1 Scottish, 4 NA English)
- Test speaker ‘bdl’ (male, NA English)
  - total of 1131 adaptation sentences available; experiments used 1%, 4%, 10%, or 100% of these
- Test sentences from various domains (conversation, novels, news, etc)

Configuration



- Synthesis model is an HSMM (explicit duration model)
- MGLSP spectral features plus noise and F0 parameters
- 5 state models, tied parameters, about 60k full context models
- No speaker adaptive training or speaker normalisation
- Synthesis uses diagonal global variance (Toda)
    - GV was calculated using only the adaptation data

Experimental setup

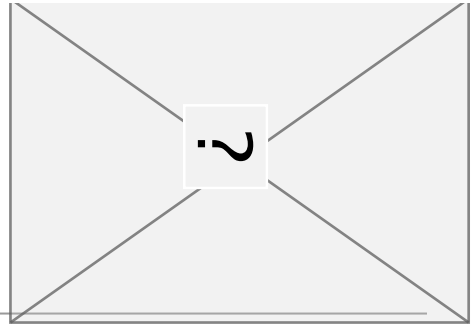


- Baseline voice for speaker ‘bdl’
  - speaker-dependent voice trained on all ‘bdl’ data

- Speaker-adapted voices for speaker ‘bdl’
  - supervised adaptation with full context labels ***vs.*** triphone labels
  - supervised (‘correct’ labels) ***vs.*** unsupervised (recognised labels)
  - adaptation scheme: MLLR mean ***vs.*** MLLR mean+var ***vs.*** CMLLR
  - adaptation data: varying amounts

- HTS version 2.1 beta was used

Baseline voice compared with natural speech



natural speech	●	●
baseline synthetic speech	●	●

Full context labels vs. triphone labels

baseline		
full context labels		
triphone labels		

supervised, CMLLR, 10% of data

Supervised vs. unsupervised

baseline	●	●
supervised	●	●
unsupervised	●	●

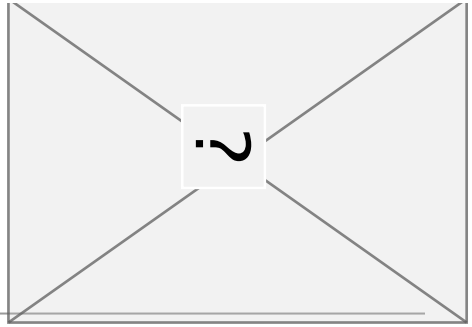
triphone labels, CMLLR, 10% of data

Amount of adaptation data

1% (12 sentences)		
10% (114 sentences)		
100% (1131 sentences)		

triphone labels, unsupervised, CMLLR

Results



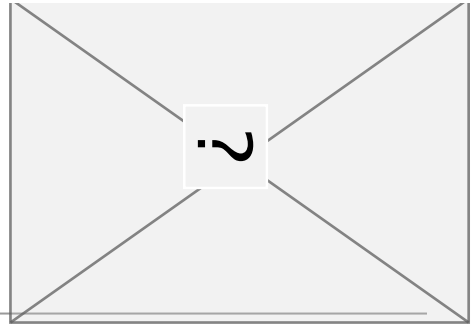
	Training data	Adaptation data	Adaptation labels	Supervised?	MOS sim	MOS news	MOS conv	WER (%)
A	all bdl	none			2.9	2.8	2.8	10.9
B	all except bdl	all bdl	full context	Y	2.4	2.7	2.5	9.3
C	all except bdl	all bdl	triphone	Y	2.2	2.4	2.7	11.5
D	all except bdl	all bdl	triphone	N	2.1	2.5	2.3	14.6
E	all except bdl	10% of bdl	full context	Y	2.5	2.7	2.7	10.6
F	all except bdl	10% of bdl	triphone	Y	2.3	2.2	2.7	13.7
G	all except bdl	10% of bdl	triphone	N	2.2	2.3	2.5	18.3
H	all except bdl	1% of bdl	full context	Y	2.0	2.4	2.5	15.8
I	all except bdl	1% of bdl	triphone	Y	2.0	2.3	2.5	15.5
J	all except bdl	1% of bdl	triphone	N	1.9	2.1	2.5	14.6

Latest results on larger dataset

supervised		
unsupervised ASR system with high WER		
unsupervised ASR system with low WER		

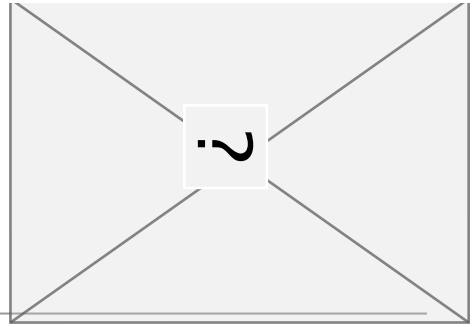
Average voice: WSJ0 SI84 (84 speakers, 7k sentences)
Adapted using 40 sentences from Nov 93 H2 task

Some conclusions



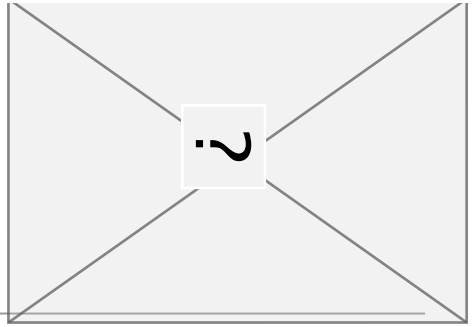
- Unsupervised adaptation is surprisingly good!
    - High accuracy ASR does not seem essential
- 40-100 sentences of adaptation data gives good performance, even for unsupervised adaptation

Summary



- Adapting full context models using only phonetic labels
- Learn transforms using triphone models, then apply to full context models
- No requirement for the full context and triphone model parameters to be linked, or to have the same tying structure etc.
- Only need a shared set of regression classes
- The method enables unsupervised adaptation
- Does not require high-accuracy recognition

Hot topics in HMM-based synthesis



- Speaker adaptation
  - unsupervised
  - cross-language
- Multi-lingual / Cross-lingual / Language-adaptive models
- Going beyond the limitations of parameter-tied HMMs
- Speech features and vocoding that are tuned to statistical modelling

End