Hidden Markov Model-based speech synthesis

Junichi Yamagishi, Korin Richmond, Simon King and many others
Centre for Speech Technology Research
University of Edinburgh, UK

www.cstr.ed.ac.uk
Note

- I did not invent HMM-based speech synthesis!

- Core idea: Tokuda (Nagoya Institute of Technology, Japan)

- Developments: many other people

- Speaker adaptation: Junichi Yamagishi (Edinburgh) and colleagues
Background
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
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

"the cat sat"
From words to linguistic specification

"the cat sat"

DET NN VB
From words to linguistic specification

"the cat sat"

DET NN VB

((the cat) sat)
From words to linguistic specification

sil dh ax k ae t s ae t sil
"the cat sat"
DET NN VB
((the cat) sat)
From words to linguistic specification

```
sil dh ax k ae t s ae t sil
"the cat sat"
DET NN VB
((the cat) sat)
```
From words to linguistic specification

sil^dh-ax+k=ae, "phrase initial", "unstressed syllable", ...

"the cat sat"

DET NN VB

((the cat) sat)
Full context models used in synthesis

\[ aa^b - l + ax = s \]
Full context models used in synthesis

```
aa^b-1+ax=s@1_3/A:1_1_3/B:0-0-3@2-1&3-3#2-2$2-3!1- .....  
```

phonetic
Full context models used in synthesis

\[ aa^{b-l+ax=s@1_3/A:1_1_3/B:0-0-3@2-1&3-3#2-2$2-3!1- \ldots \]
Example linguistic specification

pau^pau^-pau+ao=th@x_x/A:0_0_0/B:x-x-x@x-x&amp;x-x#x-x$..

pau^pau^-ao+th=er@1_2/A:0_0_0/B:1-1-2@1-2&amp;1-7#1-4$..

pau^ao^-th+er=ah@2_1/A:0_0_0/B:1-1-2@1-2&amp;1-7#1-4$..

ao^th^-er+ah=v@1_1/A:1_1_2/B:0-0-1@2-1&amp;2-6#1-4$..

th^er^-ah+v=dh@1_2/A:0_0_1/B:1-0-2@1-1&amp;3-5#1-3$..

er^ah^-v+dh=ax@2_1/A:0_0_1/B:1-0-2@1-1&amp;3-5#1-3$..

ah^v^-dh+ax=d@1_2/A:1_0_2/B:0-0-2@1-1&amp;4-4#2-3$..

v^dh^-ax+d=ey@2_1/A:1_0_2/B:0-0-2@1-1&amp;4-4#2-3$..

“Author of the ...”
From linguistic specification to speech

- Two possible methods
  - Concatenate small pieces of pre-recorded speech
  - Generate speech from a model
From linguistic specification to speech

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

- HMMs are models of sequences
  - speech signals
  - gene sequences
  - etc
HMMs

- a HMM consists of
  - sequence model: a weighted finite state network of states and transitions
  - observation model: multivariate Gaussian distribution in each state
- can generate from the model
- can also use for pattern recognition (e.g., automatic speech recognition)
HMMs are generative models
HMMs are generative models
HMMs are generative models
HMM-based speech synthesis mini-tutorial

- HMMs are used to generate sequences of speech (in a parameterised form)

- 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’
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
Generation

• Generate the most likely observation sequence from the HMM
  
  • but take the statistics of not only the static coefficients, but also the delta and delta-delta too
  
  • Maximum Likelihood Parameter Generation Algorithm
Trajectory HMMs
Trajectory HMMs
Trajectory HMMs
Trajectory HMMs
Trajectory HMMs
Trajectory HMMs
Trajectory HMMs
Trajectory HMMs
Trajectory HMMs

speech parameter

time
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 specification
  - concatenate the corresponding HMMs
  - generate from the HMM
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 specification
  - concatenate the corresponding HMMs
  - 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@1_2/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
Sparsity

• In practically all speech or language applications, sparsity is a problem

• Distribution of classes is usually long-tailed (Zipf-like)

• We also ‘create’ even more sparsity by using context-dependent models
  • thus, most models have no training data at all

• Common solution is to merge classes or contexts
  • i.e., use the same model for several classes or contexts
  • for HMMs, we call this ‘parameter tying’
Decision-tree-based clustering

Description length for $U$

\[
D(U) = \frac{1}{2} \sum_{m=1}^{M} \Gamma_m (K + K \log(2\pi) + \log |\Sigma_m|) \\
+ K M \log W + C
\]

$\Gamma_m$  State occupancy probability for node $S_m$

$K$  Dimension

$\Sigma_m$  Covariance matrix for node $S_m$

$W = \sum_{m=1}^{M} \Gamma_m$
Model parameter estimation from ‘labelled’ data

- Actually, we only have word labels for the training data

- Convert these to full linguistic specification using the front end of our text-to-speech system (text processing, pronunciation, prosody)
  
  - these labels will not exactly match the speech signal (we do a few tricks to try to make the match closer, but it’s never perfect)

- We still only know the model sequence, but no information about the state alignment

- So, we use EM (we could call this ‘semi-supervised’ learning)
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
Training, adaptation, synthesis
Training, adaptation, synthesis

speech

labels
Training, adaptation, synthesis
Training, adaptation, synthesis

Train

Average voice model

awb
clb
rms

speech

labels

...
Training, adaptation, synthesis
Training, adaptation, synthesis

Average voice model

speech → Adapt → labels

bdl → Adapt → bdl
Training, adaptation, synthesis
Training, adaptation, synthesis

- Average voice model
- Adapt
- Transforms
- Speech
- Labels
Training, adaptation, synthesis

- Speech
- BDL
- Average voice model
- Adapt
- Transforms
- Recognise
- Labels
- BDL

Effective Multilingual Interaction in Mobile Environments
Training, adaptation, synthesis

Diagram:
- Speech
- bdl
- Average voice model
- Adapt
- Transforms
- bdl
- Labels

Effective Multilingual Interaction in Mobile Environments
Training, adaptation, synthesis

Average voice model

Transforms
Training, adaptation, synthesis

Diagram:
- Test sentence labels
- Average voice model
- Synthesise
- Transforms
- Synthetic speech
Training, adaptation, synthesis

- Test sentence labels
- Average voice model
- Synthesise
- Transforms
- Synthetic speech
Training, adaptation, synthesis
Training, adaptation, synthesis
Training, adaptation, synthesis
Evaluation

- Objective measures that compare synthetic speech with a natural example (e.g., spectral distortion) have their uses, but don’t necessarily correlate with human perception

- main problem: there is more than one ‘correct answer’ in speech synthesis

- a single natural example does not capture this

- So, we mainly rely on playing examples to listeners

- opinion scores for quality & naturalness, typically on 5 point scales

- objective measures of intelligibility (type-in tests)
Intelligibility (WER), English

Word error rate for voice A (All listeners)

Word error rate for voice B (All listeners)

A natural speech
B Festival benchmark
C HTS 2005 benchmark
V HTS 2008 (aka HTS 2007’)

System
Intelligibility (WER), English

Word error rate for voice A (All listeners)

No significant difference between A, V and T

Word error rate for voice B (All listeners)

A natural speech
B Festival benchmark
C HTS 2005 benchmark
V HTS 2008 (aka HTS 2007')
Intelligibility (WER), English

Word error rate for voice A (All listeners)

Word error rate for voice B (All listeners)

No significant difference between A, V and T

No significant difference between A, C, V and T

A natural speech
B Festival benchmark
C HTS 2005 benchmark
V HTS 2008 (aka HTS 2007')
Intelligibility (WER), English

HTS is as intelligible as human speech

A natural speech
B Festival benchmark
C HTS 2005 benchmark
V HTS 2008 (aka HTS 2007’)

Word error rate for voice A (All listeners)

Word error rate for voice B (All listeners)
Recent extensions
Articulatory-controllable HMM-based speech synthesis

- can manipulate articulator positions explicitly

- ability to synthesise new phonemes, not seen in training data

- requires parallel articulatory+acoustic corpus, which we have in CSTR
Articulatory-controllable HMM-based speech synthesis

<table>
<thead>
<tr>
<th>Tongue height (cm)</th>
<th>+1.5</th>
<th>+1.0</th>
<th>+0.5</th>
<th>default</th>
<th>-0.5</th>
<th>-1.0</th>
<th>-1.5</th>
</tr>
</thead>
</table>
Articulatory-controllable HMM-based speech synthesis

<table>
<thead>
<tr>
<th>Tongue height (cm)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+1.5</td>
<td></td>
</tr>
<tr>
<td>+1.0</td>
<td></td>
</tr>
<tr>
<td>+0.5</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td>set</td>
</tr>
<tr>
<td>-0.5</td>
<td></td>
</tr>
<tr>
<td>-1.0</td>
<td></td>
</tr>
<tr>
<td>-1.5</td>
<td></td>
</tr>
</tbody>
</table>
Dirichlet process HMMs

- Fixed number of states may not be optimal
- Cross-validation, information criteria (AIC, BIC, or MDL) or variational Bayes can be used for determining the number of states
- Or use Dirichlet process (HDP-HMM or infinite HMM)
Summary

- HMM-based speech synthesis has many opportunities for using machine learning:
  - learning the model from data
    - parameters (alternatives to maximum likelihood such as minimum generation error)
    - model complexity (context clustering, number of mixture components, number of states, ...)
  - semi-supervised and unsupervised learning (labels for data are unreliable or missing)
  - adapting the model, given limited new data
  - generation algorithms