Acoustic modelling

- ASR Recognition/training issues
- HMM: recognition, training
- DTW, template matching
- Acoustic parametrization
Parametrized speech into: n features at n ms frames
Phonemic features: voicing, nasality, formants etc
- delta (delta delta)

PLP: Perceptual Liner Prediction

Overall (Mel/Bark)
Band power spectrum
- Frequency domain

Acoustic parametrization
Naive way to do speech recognition

Template matching

Comparing cepstral parameters is better
Comparing waveforms doesn't (really work)

□ Compare against each and find closest

□ Record the test utterances

□ Pre-record a bunch of words/phrases

□
Find shortest euclidean distance through table

Dynamic Time Warping
phones are different in different contexts – but getting a good example of phone is difficult
Could use phone models and chain them □

– as in „connected“ speech recognition
Could use word models and chain them □
gets very parallel very quickly

Need more general algorithm □

DTW requires an example to compare against
Thus a probability is given for each state when state is unknown:

\[
\text{Hidden Markov Models: } \sum_{\eta \in \Theta} \left( \prod_{t=1}^{m} f(t) \right) p_{\eta}(X_t) \]

Future can be predicted from (limited) past knowledge:

\[
\text{Markov Process: } \sum_{\eta \in \Theta} \left( \prod_{t=1}^{m} f(t) \right) p_{\eta}(X_t) \]

(Hidden) Markov Models
A model $n(t) = n(t-1)$

- $X \ni X \in A \ni A \ni A$ = $B$
- $S \ni S \in \alpha \ni \alpha$ = $A$
- $S \ni S \in \beta \ni \beta$ = $\beta$
- $w \in \omega \ni \omega$ = $K$
- $N \ni N \in \sigma \ni \sigma$ = $S$

Hidden Markov Models
Three fundamental questions for HMMs:

1. Given a model, how do we efficiently compute
   $P(O | \lambda)$?

2. Given observation and model $O$, which model is most probable?
   $P(\pi | O) \propto P(\lambda)$.

3. Given and a space of models, how do we find the best
   model to explain the thing $O$? In a model, what states are most likely
   best explains the observations which state sequence.
as in Chart Parsing, memoizing.

Can be done by using dynamic programming:

- Each path probability is product of each transition in state
- Find sum of all path probabilities

called decoding

\[ P(\Pi | O) = \prod_{i=1}^{n} P(O_i | \Pi, A) \]

Given observations of an observation

Finding the Probability of an Observation
However when looking for n-best we need more pointers.

- Can ignore other backpointers
  - Because we are looking for best
- Cumulated values through path
- Hold back pointer to best previous state
- At each point hold list of possible states

 maximises best sequence not each state

What is the best state sequence given the observation

Finding the Best Path
Parameter estimation

- may find local maximum
- Not globally optimal
- Repeat until convergence

- Modify probabilities to make observation path best (backwards)
- Run observation and find current probs (forwards)

- Special case of Expectation Maximization
- Baum-Welch (forward/backward)

- Use Maximum Likelihood Estimation

called training
HMM recognition

Thus can find most probable sequence through HMM states

- Probability distribution of possible phone type

Each observation (e.g. 20ms param vector):

- one for each phone type (sort of)

A bunch of HMMs

- use Viterbi to find best path
(M)p

- Use language model to provide

(M | O)p

- Use HMM to provide

Combine models:

\[
\frac{(O)p}{(M|O)p(M)p}
\]

\[
\text{Using Bayes' Law}
\]

\[
(O | M)p
\]

Find word sequence that maximizes

But not all phones are equally probable

But that's not enough
But even that's not enough.
Search space is very large

□ Prune Viterbi search:
  − Best number of paths
  − Some percentage of probability mass

□ Prune lexical trees:
  − restrict vocabulary
  − use language model
  − or grammar
Some related points

Typically negative log probabilities
- so done as logs, thus addition is used
- they get very small
Proportional to multiplies along paths
Acoustic Training

- Get more data
- Fix bugs

How do you improve speech recognition: ( )

- As close to the target domain as possible
- As much as you can get

What type (male/female): ( )

- Can take months to train
- More than 10hrs
- As much as you can get

How much data do you need: ( )
Summary

- Training of parameters by Baum-Welch
- Decoding by Viterbi

HMMs

DTW: Template comparison

CELP, delta etc

Best differentiator for phones

Acoustic parameterization