Statistical signal processing methods for speech

Richard Turner (turner@gatsby.ucl.ac.uk)

Gatsby Computational Neuroscience Unit, 19/04/2007
Introduction

Machine-audition problems include:

- Less domain-specific tasks:
  - Filling in missing data (artifact removal)
  - Denoising (first model), Deconvolution/De-reverberation
  - **Source separation**/Auditory scene analysis
  - Super-resolution, Compression ...

- More domain-specific tasks:
  - Articulatory modelling, Glottal-pulse location
  - Statistical signal processing (**time freq.** reps, PAD)
  - **Voiced/unvoiced detection**, **Pitch tracking**
  - Speech manipulation (pitch/VTL/**time-scale** etc.)

- Traditional signal processing methods are feed-forward

- Generative models: simple way of deriving complex processing algorithms
The Spectrogram

- Natural Sounds are often composed of a **small** number of sinusoidal components whose coefficients vary **slowly**.

- Good representation is: **Short-Time-Fourier-Transform** (this is **linear**),

  \[
  z_{t,\omega} = \sum_{t'} y_{t-t'} w_{t'} \exp[-i\omega t']
  \]  

  (1)

- Often interested in frequency content (not phase): take magnitude of STFT.

- Due to the sparsity, it is useful to take logs of this to give the **spectrogram**:

  \[
  x_{t,\omega} = \log |z_{t,\omega}|
  \]  

  (2)

- This is **non-linear** due to the magnitude and log operations.
The max-approximation

- STFT/Wavelet coefficients are very sparse

- When 2 independent sources are active at the same time the spectrogram is well approximated by the maximum of the source spectrograms (c.f. Jörg):

\[
x_{t,\omega}^{(1+2)} = \log |\hat{z}_{t,\omega}^{(1)} + \hat{z}_{t,\omega}^{(2)}| \sim \max \left( \log |\hat{z}_{t,\omega}^{(1)}|, \log |\hat{z}_{t,\omega}^{(2)}| \right)
\]

- Each point on the spectrogram can be thought of as being contributed entirely by one source.
Factorial MAX-VQ: Motivation

• **Goal**: group together regions of the spectrogram that belong to the same auditory object.

• **Possible cues**:
  – common onset/offset
  – common movement (sweep)
  – harmonic relationship

• Slight perverse to move from linear mixing (STFT) to non-linear occlusive mixing (spectrogram)
The Factorial MAX-VQ model

\[ p(z_m = k | \pi) = \pi_m^k \]  \hspace{1cm} (5)

\[ p(z) = \prod_{m=1}^{M} p(z_m) \] \hspace{1cm} (6)

\[ a_d = \arg \max_m (v_{z_d m}) \] \hspace{1cm} (7)

\[ p(x_d | a_d, v, \Sigma) = \text{Norm}(x_d | v_{z_a d}, \Sigma_{ad}) \] \hspace{1cm} (8)

\[ p(x | v, \Sigma, \pi) = \sum_z p(z | \pi) p(x | z, v, \Sigma) \] \hspace{1cm} (9)
Learning

• Use **MAP inference** for the $z$

• Viterbi is too slow - so use **branch and bound** techniques:

$$p(x|z_m = k) \leq \frac{1}{2}[x - v^k_m]^2 - \frac{D}{2} \log |\Sigma| - \log \pi^k_m$$ (10)

• Use this bound to quickly find MAP solutions, essentially **eliminates mixture components which can never make contributions**.

• Use an overlap and add method to use the masks to recover the original signals.
Experiments

• Train one of the mixture models on noiseless speech ($K=512$ components)

• Train a second mixture model on noise ($K = 32$ components)

• Experiment 1: Denoising - use one copy of each mixture model

• Experiment 2: Source separation - use two copies of the first model
Denoising results
Separation results
Future work

• Develop techniques for learning from only mixed/noisy data, without requiring clean, isolated examples.

• But EM requires summing over all possible settings of $z$ i.e. $K^M$.

• Therefore require approximations for large $K$ or $M$ e.g. factored variational approximations.
A Segmental HMM for Speech Waveforms: Motivation

- **Task**: isolate glottal pulses and boundaries between voiced and unvoiced speech (pitch tracking, voice speech identification, timescale modification)

- Chop up the original speech wave into natural atomic units which can be examined or manipulated in very flexible ways e.g.

- **Notation**: break waveform $s_n$ into segments: $s_{b_k} : s_{b_{k+1}-1}$

- $b_k =$ time index of the beginning of the $k^{th}$ segment
Generative model

The distribution over the boundaries is given by,

\[ b_1 = 1 \]  
\[ p(b_k | b_{k-1}) = \text{uniform}(b_{k-1} + T_{\text{min}}, b_{k-1} + T_{\text{max}}) \]  
\[ b_K = T \]  

The distribution over the segments is given by,

\[ p(s_1) = \text{Norm}(0, I\sigma_1^2) \]  
\[ p(s_k | s_{k-1}) = \text{Norm}(T(\alpha_k, \beta_k, \gamma_k)s_{k-1}, I\sigma_k^2) \]  

Subject to the constraints \( s_k(\text{start}) \& s_k(\text{end}) = 0 \)
Learning

• MAP learning for the latents

• Imagine we know the boundaries $b_{1:K}$: MAP estimates of the transformations are then **local**:

1. Find the time warping: $\alpha_k = \frac{b_{k+1} - b_k}{b_k - b_{k-1}}$
2. Decimate/interpolate $s_{k-1}$ to get $\hat{s}_{k-1}$
3. Find the optimal shift $\gamma_k$ and scale $\beta_k$ via least squares:
   \[
   \arg\min_{\beta_k, \gamma_k} = \sum_{j=1}^{b_{k+1} - b_k} \left( \beta_k \hat{s}_{k-1}(j) + \gamma_k - s_k(j) \right)^2
   \] (17)
4. Set the noise $\sigma_k^2$ using the argument of this minimisation

• Find the boundaries using dynamic programming
Experiments: Pitch tracking

mean error in the pitch estimates: 9Hz
Experiments: Vocing detection and time-scale modification

87.5% of voiced sections correctly identified
Probabilistic STFT - Motivation

• Real sounds lie on a hyper-plane in STFT space:
  – hard to modify STFTs and resynthesize sounds as modifications should be constrained to moves on the manifold.
  – hard to build generative models for STFTs as they should generate on this hyper-plane

• Alternative: build a generative model which encapsulates the assumptions that motivated the STFT:

• Natural Sounds are often composed of a small number of sinusoidal components whose coefficients vary slowly.
Probabilistic STFT - Generative model

\[
p(x_{d,t}|x_{d,t-1}) = \text{Norm}(\lambda_d x_{d,t-1}, \sigma_d^2) \quad (18)
\]

\[
p(y_t|x_t) = \text{Norm} \left[ \sum_d x_{d,t} \sin(\omega_d t + \phi_d), \sigma_y^2 \right] \quad (19)
\]

\[
p(y_t|x_t) = \text{Norm} \left[ x_t^T s_t, \sigma_y^2 \right] \quad (20)
\]

Suitable settings for the parameters,

\[
\tau_{d}^{\text{eff}} = -\frac{1}{\log \lambda_d} \geq \frac{2\pi}{\omega_d} \quad (21)
\]

\[
\sigma_{d}^2 = (1 - \lambda_d^2) \alpha_d \quad (22)
\]

\[
\alpha_d = \frac{\beta}{\omega_d} \quad (23)
\]
Probabilistic STFT - Learning

- This is a **Linear Gaussian State Space Model** with **time-varying emission weights**.

- Efficient inference proceeds via the **Kalman Smoother**
Probabilistic STFT - Results

Normal Spectrogram

Probabilistic Spectrogram
Probabilistic STFT - Extensions

• So far sparseness isn’t explicit in the model, two possible ways of remedying this

1. Make the distribution over the $x_{d,t}$ sparse
2. Make the number of sinuoids, $D$, small and learn the frequencies/phases, might then be helpful to have noisy sines $sin(\omega_d t + \phi_d + \eta_t)$

• Don’t have to use sines: Can plug any basis functions $s_t$ into $y_t = x_t^T s_t + \epsilon_t$, e.g. gamma-tones (cf Smith and Lewicki).