Probabilistic Amplitude Demodulation

Richard Turner (turner@gatsby.ucl.ac.uk)
Maneesh Sahani (maneesh@gatsby.ucl.ac.uk)

Gatsby Computational Neuroscience Unit, 12/09/2007
Motivation
Motivation
Motivation
Motivation
Motivation
Motivation: Traditional AM

![Graph showing time in seconds (s) and a red line with fluctuations]
Motivation: Traditional AM
Motivation: Traditional AM

![Graph showing time vs signal amplitude with labels '1', 'II', '×']
Motivation: Demodulate the modulator
Motivation: Demodulate the modulator
Motivation: Demodulate the modulator
Motivation: Demodulate the modulator
Motivation: Demodulation Cascade
Traditional Demodulation Algorithms: Analytic Signal

![Graph showing y and x(2) and x(1) over time/s]
Traditional Demodulation Algorithms: Analytic Signal
Traditional Demodulation Algorithms: Analytic Signal

![Graph showing "y and x(2)" and "x(1)" over time/s]

- y and x(2)
- x(1)
Traditional Demodulation Algorithms: square and lowpass
Traditional Demodulation Algorithms: square and lowpass

\[
\begin{align*}
\text{y and x}(2) \\
\text{x}(1)
\end{align*}
\]
Traditional Demodulation Algorithms: square and lowpass

![Graph showing y and x(2) over time/s with notable peaks and fluctuations]

![Graph showing x(1) with periodic oscillations and a decreasing trend over time/s]
Traditional Demodulation Algorithms are not sufficient

- **Analytic Signal Amplitude**
  - recovers a *carrier of constant variance*.
  - recovers a *modulator that may not correspond to the envelope we want*.
  - no tunable parameter

- **Square and lowpass**
  - recovers a *good-looking smooth envelope*
  - recovers a *carrier of high variance* i.e. not demodulated
  - has a tunable parameter
Advantages of the probabilistic approach

Representing a signal as:

\textbf{quickly varying carrier} × \textbf{slowly varying positive envelope}

is \textbf{fundamentally an ill posed problem we need prior information} and therefore the \textbf{probabilistic setting is the natural one}.

- Makes \textbf{assumptions explicit} allowing them to be criticised and improved

- Allows us to put \textbf{error-bars} on our inferences

- We can develop a \textbf{family of related algorithms}: cheap and cheerful/slow and Bayesian, with/without hand-tunable parameters

- \textbf{Tap into the existing theory} and methods to e.g. deal with \textbf{missing data}, \textbf{model comparison} etc.
A simple generative model for AM

\[
\begin{align*}
    z_t^{(2)} &\sim \text{Norm}(\lambda z_{t-1}^{(2)}, \sigma^2) \\
x_t^{(2)} &= f(z_t^{(2)}, a)
\end{align*}
\]
A simple generative model for AM

\[ z_t^{(2)} \sim \text{Norm}(\lambda z_{t-1}^{(2)}, \sigma^2) \]

\[ x_t^{(2)} = f(z_t^{(2)}, a) \]
A simple generative model for AM

\[ z_t^{(2)} \sim \text{Norm}(\lambda z_{t-1}^{(2)}, \sigma^2) \]

\[ x_t^{(2)} = f(z_t^{(2)}, a) \]

\[ x_t^{(1)} \sim \text{Norm}(0, 1) \]

\[ l_{\text{eff}} = -\frac{1}{\log(\lambda)} \]
A simple generative model for AM

\[ z_t^{(2)} \sim \text{Norm}(\lambda z_{t-1}^{(2)}, \sigma^2) \]

\[ x_t^{(2)} = f(z_t^{(2)}, a) \]

\[ x_t^{(1)} \sim \text{Norm}(0, 1) \]

\[ y_t = x_t^{(1)} x_t^{(2)} \]

\[ l_{\text{eff}} = -1 / \log(\lambda) \]
Learning Algorithms

The probability of everything

\[ p(y_{1:T}, x^{(1)}_{1:T}, x^{(2)}_{1:T}, \lambda | \sigma^2, a) = p(\lambda) p(y_{1:T} | x^{(1)}_{1:T}, x^{(2)}_{1:T}) p(x^{(1)}_{1:T}) p(x^{(2)}_{1:T}) \]
Learning Algorithms

Remember the Jacobian

\[ p(y_{1:T}, x^{(1)}_{1:T}, x^{(2)}_{1:T}, \lambda | \sigma^2, a) = p(\lambda)p(y_{1:T} | x^{(1)}_{1:T}, x^{(2)}_{1:T})p(x^{(1)}_{1:T})p(x^{(2)}_{1:T}) \]

\[ p(z_{1:T}) \prod_{t=1}^{T} \left| \frac{dz_t}{dx^{(2)}_t} \right| \]
Integrate out the carrier

\[
p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, \alpha) = p(\lambda) p(y_{1:T} | x_{1:T}^{(1)}, x_{1:T}^{(2)}) p(x_{1:T}^{(1)}) \underbrace{p(x_{1:T}^{(2)})}_{\ddots} p(z_{1:T}) \prod_{t=1}^{T} \left| \frac{dz_t}{dx_t^{(2)}} \right|
\]

\[
p(y_{1:T}, x_{1:T}^{(2)}, \lambda | \sigma^2, \alpha) = \int dx_{1:T}^{(1)} p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, \alpha)
\]
Learning Algorithms

Integrate out the **dynamics** ($\lambda$) (ML will not work)

\[
p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, a) = p(\lambda)p(y_{1:T} | x_{1:T}^{(1)}, x_{1:T}^{(2)})p(x_{1:T}^{(1)}) \underbrace{p(x_{1:T}^{(2)})}_{p(z_{1:T}) \prod_{t=1}^{T} \left| \frac{dz_t}{dx_t^{(2)}} \right|}
\]

\[
p(y_{1:T}, x_{1:T}^{(2)}, \lambda | \sigma^2, a) = \int dx_{1:T}^{(1)} p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, a)
\]

\[
p(y_{1:T}, x_{1:T}^{(2)} | \sigma^2, a) = \int d\lambda \int dx_{1:T}^{(1)} p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, a)
\]
Learning Algorithms: CHEAP AND CHEERFUL

Maximise over the envelope and other parameters

\[
p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, a) = p(\lambda)p(y_{1:T} | x_{1:T}^{(1)}, x_{1:T}^{(2)})p(x_{1:T}^{(1)})p(x_{1:T}^{(2)})
\]

\[
p(z_{1:T}) \prod_{t=1}^{T} \left| \frac{dz_t}{dx_t^{(2)}} \right|
\]

\[
p(y_{1:T}, x_{1:T}^{(2)}, \lambda | \sigma^2, a) = \int dx_{1:T}^{(1)} p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, a)
\]

\[
p(y_{1:T}, x_{1:T}^{(2)} | \sigma^2, a) = \int d\lambda \int dx_{1:T}^{(1)} p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, a)
\]

\[
\text{arg max } p(y_{1:T}, x_{1:T}^{(2)} | \sigma^2, a)
\]

\[
x_{1:T}^{(2)}, \sigma^2, a
\]
Sample the envelope and other parameters

\[ p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, a) = p(\lambda)p(y_{1:T} | x_{1:T}^{(1)}, x_{1:T}^{(2)})p(x_{1:T}^{(1)})p(x_{1:T}^{(2)}) \]

\[ p(z_{1:T}) \prod_{t=1}^{T} \left| \frac{dz_t}{dx_{t}^{(2)}} \right| \]

\[ p(y_{1:T}, x_{1:T}^{(2)}, \lambda | \sigma^2, a) = \int dx_{1:T}^{(1)} p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, a) \]

\[ p(y_{1:T}, x_{1:T}^{(2)} | \sigma^2, a) = \int d\lambda \int dx_{1:T}^{(1)} p(y_{1:T}, x_{1:T}^{(1)}, x_{1:T}^{(2)}, \lambda | \sigma^2, a) \]

\[ p(x_{1:T}^{(2)} | y_{1:T}, \sigma^2, a) = \frac{1}{p(y_{1:T} | \sigma^2, a)p(y_{1:T} | \sigma^2, a)} p(y_{1:T}, x_{1:T}^{(2)} | \sigma^2, a) \]
Results: Vanilla Probabilistic Amplitude Demodulation (PAD)
Results: PAD tuning parameters for phonemes
Results: PAD tuning parameters for phonemes

![Graph showing PAD tuning parameters for phonemes]
Results: PAD tuning parameters for phonemes
Results: PAD tuning parameters for pitch

![Graph showing y and x(2) trends over time/s](image)

![Graph showing x(1) trends over time/s](image)
Results: PAD tuning parameters for pitch

The diagram shows the time evolution of variables $y$ and $x(2)$, and $x(1)$ over time in seconds (s). The data points are plotted against time to visualize the behavior of these variables over a specified period.
Results: PAD tuning parameters for pitch
New Model: Demodulation Cascade
Results: Demodulation Cascade

![Graph showing time series data for y and x](image-url)
Results: Demodulation Cascade

![Graph showing demodulation cascade results with time axis labeled as 'time /s'.]
Mode: sentences
Mean and Error-bars: Hamiltonian MCMC

![Graph showing time series data for y and x(2) with error bars, and x(1) with time scale]
Mode: phonemes
Mode: cascade
Error-bars: Hamiltonian MCMC

![Graph showing error-bars for Hamiltonian MCMC](image_url)
Part 2: Current Work
Modeling the fine temporal structure
Modeling the fine temporal structure
Modeling the fine temporal structure
Modeling the fine temporal structure
Modeling the fine temporal structure
Modeling the fine temporal structure
Frequency Modulation and Instantaneous Frequency

\[ x_t^{(2)} \]
Frequency Modulation and Instantaneous Frequency

\[ x(t) \]

\[ \omega(t) \]

\[ \chi(t) \]

\[ \chi^{(2)}(t) \]
Frequency Modulation and Instantaneous Frequency

\[ x(t) = \sin(\omega_0 + \frac{\pi}{2} t) \]

\[ \omega(t) = \omega_0 + \frac{\pi}{2} t \]

Graphs showing the modulation and instantaneous frequency. The left graph displays \( \omega(t) \) and \( \omega_0 \), while the right graphs show \( x(t) \) with different modulation levels.
Frequency Modulation and Instantaneous Frequency

\[ \omega_t = \frac{2\pi}{\omega_0} \cdot t \]

\[ x(t) = \sin(2\pi f_0 t) \]

\[ y(t) = \sin(2\pi f_0 t + \phi(t)) \]

\[ \phi(t) = \int \omega_f(t) \, dt \]

\[ \omega_f(t) = \frac{\Delta f}{2} \cdot t \]

\[ x(2) = \sin(2\pi \cdot 2 f_0 t) \]

\[ x(1) = \sin(2\pi f_0 t) \]

\[ y(t) = \sin(2\pi f_0 t + \phi(t)) \]

\[ \phi(t) = \int \omega_f(t) \, dt \]

\[ \omega_f(t) = \frac{\Delta f}{2} \cdot t \]

\[ x(2) = \sin(2\pi \cdot 2 f_0 t) \]

\[ x(1) = \sin(2\pi f_0 t) \]

\[ y(t) = \sin(2\pi f_0 t + \phi(t)) \]

\[ \phi(t) = \int \omega_f(t) \, dt \]

\[ \omega_f(t) = \frac{\Delta f}{2} \cdot t \]
Proof of concept
Proof of concept

![Graph showing oscillatory behavior with frequency and time plots.](image)
Proof of concept

![Graph showing a sine wave and frequency over time]

- **Y-axis:** Amplitude
- **X-axis:** Time (s)
- **ω/2π (Hz):** Frequency

The graph illustrates a proof of concept for a system's behavior over time, showing how amplitude and frequency change with respect to time.
Proof of concept

![Graph showing y and \(\omega/2\pi\) vs. time]
Probabilistic Short Time Fourier Transform
Probabilistic Short Time Fourier Transform
Probabilistic Short Time Fourier Transform
Summary

- **AM** is an important **long-range structure** present in many natural sounds.
- Developed a **family of algorithms** for **amplitude demodulation**.
- Proposed the **demodulation cascade** as a useful representation of natural sounds.
- Current Work:
  - Extending to simultaneous **frequency and amplitude demodulation**.
  - Combining mixtures of these to make **Probabilistic STFT** representation.
- Many **traditional signal processing algorithms** address essentially ill-posed problems are therefore **often best viewed as problems of Bayesian inference**.
AM and FM

- PAD puts a **terrible** prior over the pitch and formant structure: **white-noise**

- An alternative is motivated by **Instantaneous Frequency**;

\[
y_t = x_t^{(2)} x_t^{(1)} = x_t^{(2)} \cos(\theta_t)
\]

\[
\omega_t = \theta_t - \theta_{t-1}
\]

- Place priors over the instantaneous frequency e.g. \( \omega_t \) is the mean frequency plus a slowly moving perturbation,

\[
\omega_t = \langle \omega \rangle + z_t
\]

\[
\theta_t = \langle \omega \rangle t + \sum_{t'=1}^{t} z_{t'}
\]
A simple generative model for FM

\[ p(x_t^{(2)}|x_{t-1}^{(2)}) = \text{Norm}\left[ \lambda x_{t-1}^{(2)}, \sigma_x^2 (1 - \lambda^2) \right] \]
A simple generative model for FM

\[
p(x_t^{(2)} | x_{t-1}^{(2)}) = \text{Norm} \left[ \lambda x_{t-1}^{(2)}, \sigma_x^2 (1 - \lambda^2) \right]
\]

\[
p(z_t | z_{t-1}) = \text{Norm} \left[ \lambda z_{t-1}, \sigma_z^2 (1 - \lambda^2) \right]
\]

\[
\theta_t = \sum_{t'} \omega_{t'} = \langle \omega \rangle_t + \sum_{t=t'} z_{t'}
\]
A simple generative model for FM

\[ p(x_t^{(2)}|x_{t-1}^{(2)}) = \text{Norm} \left[ \lambda x_{t-1}^{(2)}, \sigma_x^2(1 - \lambda^2) \right] \]

\[ p(z_t|z_{t-1}) = \text{Norm} \left[ \lambda z_{t-1}, \sigma_z^2(1 - \lambda^2) \right] \]

\[ \theta_t = \sum_{t'} \omega_{t'} = \langle \omega \rangle t + \sum_{t=t'} z_{t'} \]

\[ p(y_t|x_t^{(2)}, z_t) = \text{Norm} \left[ x_t^{(2)} \cos(\theta_t), \sigma_y^2 \right] \]
Learning: Variational EM

- Imagine we know the phase at each time point ($z_{1:T}$)

- Then updates for $x_{1:T}^{(2)}$ are simple: Kalman Smoother with time varying weights

  $$p(y_t|x_t^{(2)}, z_t) = \text{Norm} \left[ x_t^{(2)} c_t, \sigma_y^2 \right]$$

- How do we learn the phases?
  - either via MAP
  - or via sampling to get sufficient stats $\langle c_t \rangle$ and $\langle c_t^2 \rangle$

- Causal annealing really helps to avoid local minima induced by the $2\pi$ degeneracy in phase
Part 3: A generative model for speech
Probabilistic Amplitude Demodulation

\[ x^{(3)}_t = x^{(2)}_t x^{(1)}_t y_t' \]
Probabilistic Short Time Fourier Transform
Combination: Generative Model for Speech
New modulation cascade model

\[ x^{(3)}_t \]

\[ x^{(2)}_{1:4,t} \]

\[ y_{1:T} \]

time /s
New modulation cascade model

\[ x_{1:4,t} \]

\[ y_{1:T} \]
New modulation cascade model
New modulation cascade model
New modulation cascade model
Results
Modulation Cascade Process Results: Sentence, K1 = 24, K2 = 6, D = 200
Results