Lecture 7
Frequency Domain Methods

Dennis Sun
Stats 253

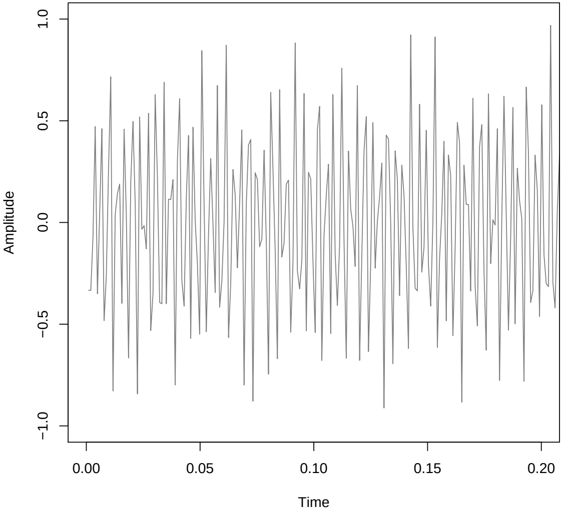
July 14, 2014

Outline of Lecture

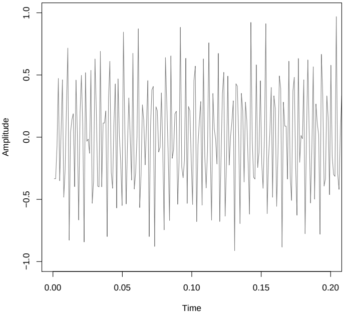
1. The Frequency Domain
2. (Discrete) Fourier Transform
3. Spectral Analysis
4. Projects

Where are we?

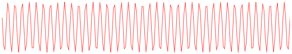
A Time Series



A Time Series

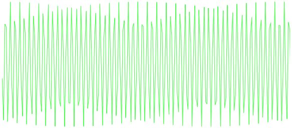


$=$



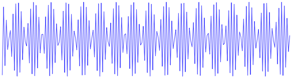
$$.2\cos(2\pi \cdot 196t)$$

$+$



$$.5\cos(2\pi \cdot 294t)$$

$+$



$$.3\cos(2\pi \cdot 470t)$$

Recovering the Weights

Suppose we knew that the only frequencies in the sound were 196, 294, and 470 Hz and we wanted to know the weights.

$$\begin{pmatrix} y(t_1) \\ \vdots \\ y(t_n) \end{pmatrix} = \begin{pmatrix} \cos(2\pi \cdot 196 t_1) \\ \vdots \\ \cos(2\pi \cdot 196 t_n) \end{pmatrix} \lambda_1 + \begin{pmatrix} \cos(2\pi \cdot 294 t_1) \\ \vdots \\ \cos(2\pi \cdot 294 t_1) \end{pmatrix} \lambda_2 + \begin{pmatrix} \cos(2\pi \cdot 470 t_1) \\ \vdots \\ \cos(2\pi \cdot 470 t_1) \end{pmatrix} \lambda_3$$

This is equivalent to

$$\begin{pmatrix} y(t_1) \\ \vdots \\ y(t_n) \end{pmatrix} = \begin{pmatrix} \cos(2\pi \cdot 196 t_1) & \cos(2\pi \cdot 294 t_1) & \cos(2\pi \cdot 490 t_1) \\ \vdots & \vdots & \vdots \\ \cos(2\pi \cdot 196 t_n) & \cos(2\pi \cdot 294 t_n) & \cos(2\pi \cdot 490 t_n) \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix}$$

Can write this as $\boldsymbol{y} = A\boldsymbol{\lambda}$ and solve by least squares.

Harmonic Regression

This is called **harmonic regression**.

```
Call:
lm(formula = y ~ cos.196 + cos.294 + cos.470 - 1)

Coefficients:
cos.196  cos.294  cos.470
    0.2      0.5      0.3
```

Transforming to the Frequency Domain

$$\boldsymbol{y} = A\boldsymbol{\lambda}$$

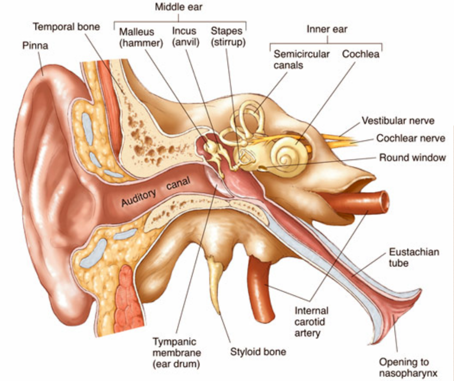
- What if we don't know the frequencies?
- We can try to include as many sinusoids $\cos(f_k t)$ in A as possible.
- Since $\boldsymbol{y}$ contains n observations, A can be at most $n \times n$.
- Now A is full rank, so it is invertible and we also have

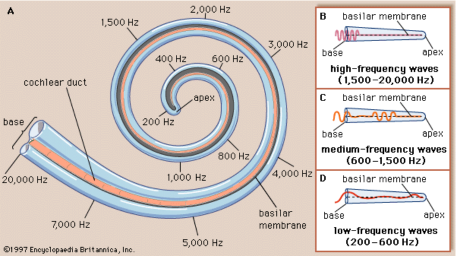
$$\boldsymbol{\lambda} = A^{-1}\boldsymbol{y}$$

- $\boldsymbol{\lambda}$ is an equivalent representation of the signal in the **frequency domain**. ($\boldsymbol{y}$ is the signal in the **time domain**.)
- A is a transform that maps $\boldsymbol{\lambda} \to \boldsymbol{y}$. A^{-1} is the inverse transform.

Why is the frequency domain relevant for sound?

Because the ear is a frequency domain analyzer!





Where are we?

1. The Frequency Domain
2. (Discrete) Fourier Transform
3. Spectral Analysis
4. Projects

Why the Fourier Transform

- In general, calculating $\boldsymbol{\lambda} = A^{-1}\boldsymbol{y}$ requires $O(n^2)$ operations
- For special choices of A, it's possible to do it in $O(n \log n)$ operations.
- For example, we might choose A to contain the complex exponentials

$$A = \begin{pmatrix} e^{jf_1 t_1} & \dots & e^{jf_n t_1} \\ \vdots & & \vdots \\ e^{jf_1 t_n} & \dots & e^{jf_n t_n} \end{pmatrix}, \quad j = \sqrt{-1}.$$

  This is called the **Discrete Fourier Transform** (DFT).
- *Note:* $e^{jf_k t_i} = \cos(f_k t_i) + j \sin(f_k t_i)$
- The fast algorithm for computing the DFT is called the **Fast Fourier Transform** (FFT).

The Fourier Transform

$$\text{DFT}: \quad \lambda(f_k) = \frac{1}{n}\sum_{i=1}^{n} y(t_i)e^{-jf_k t_i} \qquad \boldsymbol{\lambda} = A^{-1}\boldsymbol{y}$$

$$\text{Inverse DFT}: \quad y(t_i) = \sum_{k=1}^{n} \lambda(f_k)e^{jf_k t_i} \qquad \boldsymbol{y} = A\boldsymbol{\lambda}$$

- The frequencies f_k and times t_i depend on the **sampling rate** f_s.
- For example, CDs sample at 44.1 kHz, so $t_1 = 0$, $t_2 = 1/44100$.
- $t_i = i/f_s$, $f_k = f_s \cdot 2\pi k/n$
- The "unitless" form of the DFT might be easier to work with conceptually, but you have to add the units back in at the end:

$$\text{DFT}: \quad \lambda_k = \frac{1}{n}\sum_{i=1}^{n} y_i e^{-j2\pi ki/n}$$

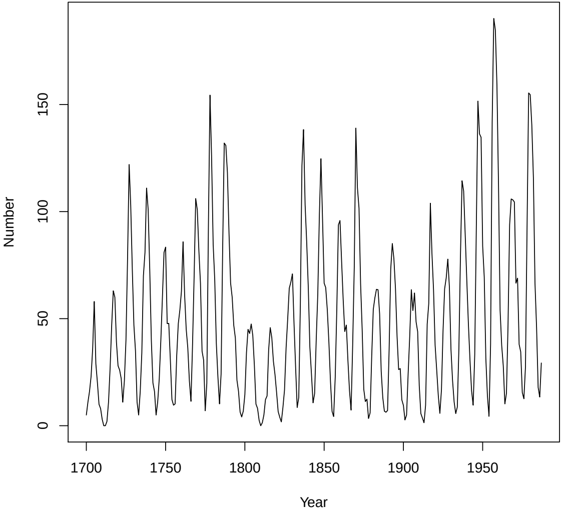
$$\text{Inverse DFT}: \quad y_i = \sum_{k=1}^{n} \lambda_k e^{j2\pi ki/n}$$

The Fourier Transform

- **Remember:** The A matrix contains complex numbers. So the frequency domain representation $\boldsymbol{\lambda} = A^{-1}\boldsymbol{y}$ is also complex-valued.
- For interpretability, we often look at the magnitudes. If $\lambda_k = a_k + jb_k$, then
$$|\lambda_k| = \sqrt{a_k^2 + b_k^2}.$$
- Note that $\boldsymbol{y} = A\boldsymbol{\lambda}$ must be real-valued. This imposes constraints on $\boldsymbol{\lambda}$.
- Let's hack around in R: `abs(fft(y))`

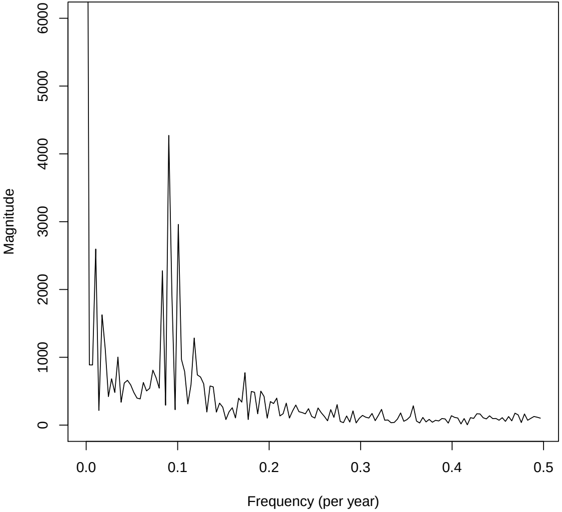
Application to Seasonality Estimation

Wolfer sunspot data



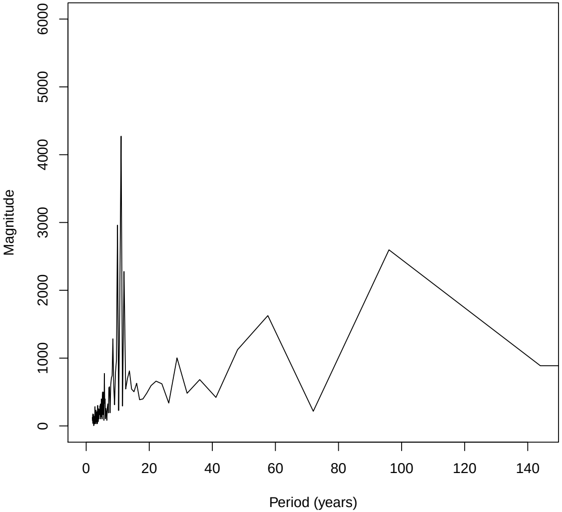
Application to Seasonality Estimation

Wolfer sunspot data: `plot(abs(fft(sunspot)))`



Application to Seasonality Estimation

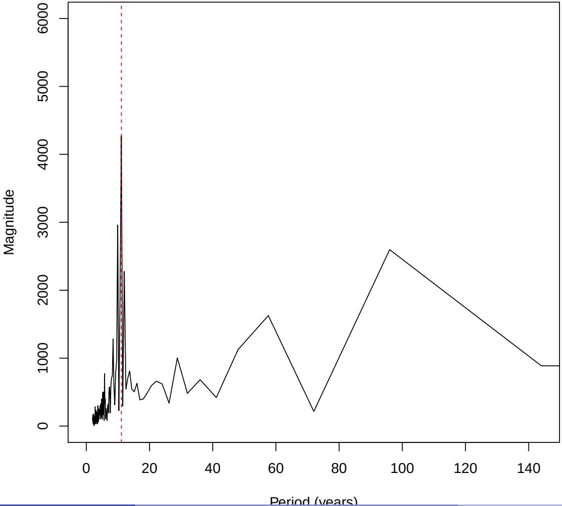
Wolfer sunspot data: Plot against period $p = 1/f$ instead of frequency.



Application to Seasonality Estimation

Wolfer sunspot data:

```
p <- 1 / ((which(lambda == max(lambda[2:n]))-1)/n)
```



Summary

- We now have a new representation of data, which is sometimes more enlightening than the time domain.
- We obtain this by taking the DFT and looking at the *magnitudes* of the resulting coefficients.
- We use the DFT (as opposed to some other transform) because it can be computed efficiently using the FFT.
- There is a 2D version of the DFT for spatial data.

Where are we?

Random Processes

- We've been using the Fourier transform to decompose a function (i.e., the trend term in $y_t = \mu_t + \epsilon_t$).
- Can we use it to study a random process ϵ_t?
- Let's do some R simulations.

Power Spectral Density

- One way to obtain a stationary random process is to take a linear combination of sinusoids, i.e.,

$$y(t) = \sum_{k=1}^{n} \lambda(f_k) e^{j f_k t}$$

  where $\lambda(f_k)$ are independent $N(0, s(f_k))$.
- The autocorrelation function is

$$C(h) = \mathrm{E}[y(t+h)\overline{y(t)}] = \mathrm{E}\left[\left(\sum_{k=1}^{n} \lambda(f_k) e^{j f_k (t+h)}\right)\left(\sum_{\ell=1}^{n} \overline{\lambda(f_\ell)} e^{-j f_\ell t}\right)\right]$$

$$= \sum_{k=1}^{n}\sum_{\ell=1}^{n} \mathrm{E}(\lambda(f_k)\overline{\lambda(f_\ell)}) e^{j(f_k - f_\ell)t} e^{j f_k h} = \sum_{k=1}^{n} \underbrace{\mathrm{E}(\lambda^2(f_k))}_{s(f_k)} e^{j f_k h}$$

- The autocorrelation function $C(h)$ is a Fourier pair with $s(f)$, which is called the **power spectral density**.

Spectral Representation Theorem

The **spectral representation theorem** says that *all* stationary processes have this representation (at least in continuous time):

$$y(t) = \int e^{jft} d\Lambda(f)$$

where Λ is a random zero-mean process with independent increments.

The **power spectral density** s is the Fourier transform of the autocorrelation function.

$$s(f) = \int C(h) e^{-jfh}\, dh$$

Spectral Density Estimation

How do we estimate $s(f)$ given samples $y(t_i)$, $i = 1, ..., n$?

- **Sample PSD:** Calculate autocorrelations and take Fourier transform.

$$\hat{s}(f) = \frac{1}{n} \sum_{h=-n+1}^{n-1} \hat{C}(h) e^{-jfh}$$

where $\hat{C}(h) = \frac{1}{n - |h|} \sum_i y_i y_{i+h}$.

Spectral Density Estimation

How do we estimate $s(f)$ given samples $y(t_i)$, $i = 1, ..., n$?

- **Periodogram:** Take Fourier transform and calculate magnitudes squared.

$$\hat{p}(f) = \left| \frac{1}{n} \sum_{i=1}^{n} y_i e^{-jft_i} \right|^2 = \left(\frac{1}{n} \sum_{i=1}^{n} y_i e^{-jft_i} \right) \overline{\left(\frac{1}{n} \sum_{m=1}^{n} y_m e^{-jft_m} \right)}$$
$$= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{n} \sum_{m=1}^{n} y_i y_m e^{-jf(i-m)/f_s}$$
$$= \frac{1}{n} \sum_{h=-n+1}^{n-1} \underbrace{\left[\frac{1}{n} \sum_{m} y_{m+h} y_m \right]}_{\frac{(n-|h|)}{n}\hat{C}(h)} e^{-jfh/f_s}$$

- **Theorem:** As $n \to \infty$, $\hat{s}(f), \hat{p}(f) \Rightarrow s(f)\chi_2^2/2$.
- So neither $\hat{s}$ or $\hat{p}$ estimates $s(f)$ consistently.

Periodogram Smoothing

Very simple solution: **smooth the periodogram**.

Let $N_f = \{k : |f_k - f| \leq B\}$ be all DFT frequencies that are within a bandwidth B of f. Then:

$$\hat{p}_{smooth}(f) = \frac{1}{|N_f|} \sum_{k \in N_f} \hat{p}(f_k)$$

Where are we?

Project Proposals

- Project proposals are due Friday.
- **Remember:** Goal is to do something useful.
- Please make clear in your project proposal what you plan to do with this project (i.e., publish a paper, release an R package, etc.).
- I will send out an (anonymous) survey about the class. When you complete that survey, you will see a link to a form to submit the project proposal.

Project Ideas

- Covariance modeling with kriging that exploits sparse matrix structure.
- Using spectral density estimation to estimate ARMA parameters.
- Next class: music applications

Administrivia

- Graded Homework 1's will be returned now. Solutions posted.
- Please turn in Homework 2.
- Homework 3 will be posted in a few hours. This one is a prediction competition using kriging methods!
- Don't forget about the project proposal.