CS 6604: Data Mining Large Networks and Time-series

B. Aditya Prakash

Lecture #12: Time Series Mining
Patterns
Patterns

More Data
Patterns

Anomaly
Patterns

Anomaly

Extrapolation

X

Y
Patterns

Imputation

Extrapolation

Anomaly
FOURIER TRANSFORM

Based on slides by Christos Faloutsos.
DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)
Introduction

Goal: given a signal (eg., sales over time and/or space)

Find: patterns and/or compress

lynx caught per year

year

count
What does DFT do?

A: highlights the periodicities
Why should we care?

A: several real sequences are periodic
Q: Such as?
Why should we care?

A: several real sequences are periodic

Q: Such as?

A:

- sales patterns follow seasons;
- economy follows 50-year cycle
- temperature follows daily and yearly cycles

Many real signals follow (multiple) cycles
Why should we care?

For example: human voice!

- Frequency analyzer
- Speaker identification
- Impulses/noise -> flat spectrum
- High pitch -> high frequency
DFT and stocks

- Dow Jones Industrial index, 6/18/2001-12/21/2001
DFT and stocks

- Dow Jones Industrial index, 6/18/2001-12/21/2001
- just 3 DFT coefficients give very good approximation

Log(ampl)

freq
DFT: definition

- Discrete Fourier Transform (n-point):

\[
X_f = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t * \exp(-j2\pi ft/n)
\]

\(j = \sqrt{-1}\)

inverse DFT

\[
x_t = \frac{1}{\sqrt{n}} \sum_{f=0}^{n-1} X_f * \exp(+j2\pi ft/n)
\]
How does it work?

Decomposes signal to a sum of sine (and cosine) waves.

Q: How to assess ‘similarity’ of \( x \) with a wave?

\[ x = \{ x_0, x_1, \ldots, x_{n-1} \} \]
How does it work?

A: consider the waves with frequency 0, 1, ...; use the inner-product (~cosine similarity)
How does it work?

A: consider the waves with frequency 0, 1, ...; use the inner-product (\textasciitilde\text{cosine similarity})
How does it work?

‘basis’ functions

sine, freq = 1

sine, freq = 2

cosine, f=1

cosine, f=2
How does it work?

- Basis functions are actually n-dim vectors, **orthogonal** to each other
- ‘similarity’ of $x$ with each of them: inner product
- DFT: $\sim$ all the similarities of $x$ with the basis functions
How does it work?

Since \( e^{jf} = \cos(f) + j \sin(f) \)

\( (j = \sqrt{-1}) \),

we finally have:
DFT: definition

- Discrete Fourier Transform (n-point):

\[ X_f = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t \cdot \exp(-j \frac{2\pi}{n} ft) \]

\( (j = \sqrt{-1}) \)

\[ x_t = \frac{1}{\sqrt{n}} \sum_{f=0}^{n-1} X_f \cdot \exp(+j \frac{2\pi}{n} ft) \]

inverse DFT
DFT: definition

- **Good news:** Available in all symbolic math packages, eg., in ‘mathematica’

  \[
  x = [1,2,1,2];
  X = \text{Fourier}[x];
  \text{Plot}[\text{Abs}[X]];\]
DFT: definition

(variations:

- $1/n$ instead of $1/\sqrt{n}$
- $\exp(-...)$ instead of $\exp(+...)$
)

DFT: definition

Observations:

- $X_f$: are complex numbers except $X_0$, who is real
- $\text{Im } (X_f)$: ~ amplitude of sine wave of frequency $f$
- $\text{Re } (X_f)$: ~ amplitude of cosine wave of frequency $f$
- $x$: is the sum of the above sine/cosine waves
DFT: definition

Observation - SYMMETRY property:

\[ X_f = (X_{n-f})^* \]

(“*”: complex conjugate: \((a + b j)^* = a - b j\))
DFT: definition

Definitions

- $A_f = |X_f| :$ amplitude of frequency $f$
- $|X_f|^2 = \text{Re}(X_f)^2 + \text{Im}(X_f)^2 :$ energy of frequency $f$
- phase $\phi_f$ at frequency $f$
DFT: definition

Amplitude spectrum: $|X_f| \text{ vs } f (f=0, 1, \ldots, n-1)$

SYMMETRIC (Thus, we plot the first half only)
DFT: definition

Phase spectrum $|\phi_f|$ vs $f$ ($f=0, 1, ... n-1$):

Anti-symmetric

(Rarely used)
DFT: examples

flat

Amplitude

time

freq
DFT: examples

Low frequency sinusoid

time

freq
**DFT: examples**

- **Sinusoid - symmetry property:** \( X_f = X^*_{n-f} \)

![Graph showing sinusoid and its frequency representation]
DFT: examples

- Higher freq. sinusoid
DFT: examples

examples

\[ \text{Graph 1} = \text{Graph 2} + \text{Graph 3} \]
DFT: examples

Examples

Ampl.

Freq.
DFT: Amplitude spectrum

Amplitude: \[ A_f^2 = \Re^2 (X_f) + \Im^2 (X_f) \]
DFT: Amplitude spectrum

**Count**

**Ampl.**

- freq=0
- freq=12

**Year**

**Freq.**
DFT: Amplitude spectrum

count

year

Ampl.

Freq.

actual  mean  mean+freq12

freq=0

freq=12
DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?

![Graphs showing DFT amplitude spectrum with actual, mean, and mean+freq12 lines.](image)
DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?
- A1: compression
- A2: pattern discovery
- (A3: forecasting)
DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?
- A1: (lossy) compression
- A2: pattern discovery
DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?
- A1: (lossy) compression
- A2: pattern discovery
DFT: Amplitude spectrum

- Let’s see it in action!
- plain sine
- phase shift
- two sine waves
- the ‘chirp’ function
Plain sine

Number of samples: 256
Sampling rate: 8000 samples / s
Signal waveform expression: \( \sin(2000\pi t) \)

[Plot signal] [Plot spectrum]
Plain sine

Number of samples: 256
Sampling rate: 8000 samples/s
Signal waveform expression: sin(2000*π*t)
Plain sine – phase shift

Number of samples: 256
Sampling rate: 8000 samples / s
Signal waveform expression: \( \sin(2000\pi t + 1.2) \)
Plain sine – phase shift

Number of samples: 256
Sampling rate: 8000 samples / s
Signal waveform expression: \( \sin(2000 \pi t + 1.2) \)
Plain sine

Number of samples: 256
Sampling rate: 8000 samples/s
Signal waveform expression: $\sin(2000\pi t)$
Two sines

Number of samples: 256
Sampling rate: 8000 samples / s
Signal waveform expression: \(\sin(2000\pi t) + 2\cos(3000\pi t + 0.5)\)

Plot signal  Plot spectrum
Two sines

Number of samples: 256
Sampling rate: 8000 samples/s
Signal waveform expression: \( \sin(2000\pi t) + 2\cos(3000\pi t + 0.5) \)
Chirp

Number of samples: 256
Sampling rate: 8000 samples / s
Signal waveform expression: $\sin(25000\pi t^2 t)$

Plot signal Plot spectrum
Chirp

Number of samples: 256
Sampling rate: 8000 samples / s
Signal waveform expression: $\sin(25000\pi t^2 t)$

Plot signal  Plot spectrum
DFT: Parseval’s theorem

\[ \sum (x_t^2) = \sum (|X_f|^2) \]

I.e., DFT preserves the ‘energy’
or, alternatively: it does an axis rotation:

\[ x = \{x_0, x_1\} \]
DFT: Parseval’s theorem

\[ \sum \left( x_t^2 \right) = \sum \left( |X_f|^2 \right) \]

I.e., DFT preserves the ‘energy’

or, alternatively: it does an axis rotation:

\[ x = \{x_0, x_1\} \]
DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)
Arithmetic examples

- Impulse function: \( x = \{ 0, 1, 0, 0 \} \ (n = 4) \)
- \( X_0 = ? \)
Arithmetic examples

- Impulse function: \( x = \{0, 1, 0, 0\} \) \((n = 4)\)
- \( X_0 =? \)
- \( A: X_0 = 1/\sqrt{4} \times 1 \times \exp(-j 2 \pi 0 / n) = 1/2 \)
- \( X_1 =? \)
- \( X_2 =? \)
- \( X_3 =? \)
Arithmetic examples

- Impulse function: \( x = \{ 0, 1, 0, 0 \} \) \((n = 4)\)
- \( X_0 = ? \)
- A: \( X_0 = \frac{1}{\sqrt{4}} * 1 * \exp(-j \frac{2 \pi 0}{n}) = \frac{1}{2} \)
- \( X_1 = -\frac{1}{2} j \)
- \( X_2 = -\frac{1}{2} \)
- \( X_3 = +\frac{1}{2} j \)
- Q: does the ‘symmetry’ property hold?
Arithmetic examples

- Impulse function: \( x = \{ 0, 1, 0, 0 \} \) \((n = 4)\)
- \( X_0 = ? \)
- A: \( X_0 = \frac{1}{\sqrt{4}} \times 1 \times \exp(-j \frac{2\pi 0}{n}) = \frac{1}{2} \)
- \( X_1 = -\frac{1}{2} j \)
- \( X_2 = -\frac{1}{2} \)
- \( X_3 = +\frac{1}{2} j \)
- Q: does the ‘symmetry’ property hold?
- A: Yes (of course)
Arithmetic examples

- Impulse function: \( x = \{ 0, 1, 0, 0 \} \) \((n = 4)\)
- \( X_0 = ? \)
- A: \( X_0 = \frac{1}{\sqrt{4}} \times 1 \times \exp(-j \frac{2 \pi 0}{n}) = \frac{1}{2} \)
- \( X_1 = -\frac{1}{2} j \)
- \( X_2 = -\frac{1}{2} \)
- \( X_3 = +\frac{1}{2} j \)
- Q: check Parseval’s theorem
Arithmetic examples

- Impulse function: \( x = \{ 0, 1, 0, 0 \} \) \((n = 4)\)
- \( X_0 = ? \)
- A: \( X_0 = \frac{1}{\sqrt{4}} \times 1 \times \exp(\frac{-j 2 \pi 0}{n}) = \frac{1}{2} \)
- \( X_1 = -\frac{1}{2} j \)
- \( X_2 = -\frac{1}{2} \)
- \( X_3 = +\frac{1}{2} j \)
- Q: (Amplitude) spectrum?
Arithmetic examples

- Impulse function: $x = \{0, 1, 0, 0\} \ (n = 4)$
- $X_0 = ?$
- A: $X_0 = \frac{1}{\sqrt{4}} \times 1 \times \exp(-j \frac{2\pi 0}{n}) = \frac{1}{2}$
- $X_1 = -\frac{1}{2}j$
- $X_2 = -1/2$
- $X_3 = +\frac{1}{2}j$
- Q: (Amplitude) spectrum?
- A: FLAT!
Arithmetic examples

- Q: What does this mean?
Arithmetic examples

- Q: What does this mean?
- A: All frequencies are equally important ->
  - we need $n$ numbers in the frequency domain to represent just one non-zero number in the time domain!
  - “frequency leak”
DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)
DFT of ‘step’ function:
\[ x = \{ 0, 0, \ldots, 0, 1, 1, \ldots 1 \} \]
Observations

- DFT of ‘step’ function:
  \[ x = \{ 0, 0, ..., 0, 1, 1, ... 1 \} \]
Observations

- DFT of ‘step’ function:
  \[ x = \{ 0, 0, \ldots, 0, 1, 1, \ldots 1 \} \]

![Graph showing the DFT of a step function with frequency components]

- For \( f = 0 \) and \( f = 1 \)
Observations

- DFT of ‘step’ function:
  \[ x = \{ 0, 0, ..., 0, 1, 1, ... 1 \} \]

- the more frequencies,
  the better the approx.

- ‘ringing’ becomes worse

- reason: discontinuities; trends
Observations

- Ringing for trends: because DFT ‘sub-consciously’ replicates the signal

![Graph showing a trend over time]
Observations

- Ringing for trends: because DFT ‘sub-consciously’ replicates the signal
Observations

- Ringing for trends: because DFT ‘sub-consciously’ replicates the signal
original
DC and 1st
DC and 1st

And 2nd
DC and 1st
And 2nd
And 3rd
DC and 1st
And 2nd
And 3rd
And 4th
Observations

- Q: DFT of a sinusoid, eg.
  \[ x_t = 3 \sin\left(\frac{2\pi}{4} t\right) \]
  \((t = 0, \ldots, 3)\)
- Q: \(X_0 = ?\)
- Q: \(X_1 = ?\)
- Q: \(X_2 = ?\)
- Q: \(X_3 = ?\)
Observations

- Q: DFT of a sinusoid, eg.
  \[ x_t = 3 \sin\left( \frac{2 \pi}{4} t \right) \]
  \((t = 0, \ldots, 3)\)

- Q: \(X_0 = 0\)
- Q: \(X_1 = -3j\)
- Q: \(X_2 = 0\)
- Q: \(X_3 = 3j\)

• check ‘symmetry’
• check Parseval
Observations

- Q: DFT of a sinusoid, eg.
  \[ x_t = 3 \sin \left( \frac{2\pi}{4} t \right) \]
  
  \(t = 0, \ldots, 3\)

- Q: \(X_0 = 0\)
- Q: \(X_1 = -3j\)
- Q: \(X_2 = 0\)
- Q: \(X_3 = 3j\)

- Does this make sense?

\[ A_f \]

\[ 0 \ 1 \ 2 \ f \]
Property

- Shifting $x$ in time does NOT change the amplitude spectrum
- eg., $x = \{0 \ 0 \ 0 \ 1\}$ and $x' = \{0 \ 1 \ 0 \ 0\}$: same (flat) amplitude spectrum
- (only the phase spectrum changes)
- Useful property when we search for patterns that may ‘slide’
DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
- DCT
- 2-d DFT
- Fast Fourier Transform (FFT)
DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)
What is the complexity of DFT?

\[ X_f = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t \exp(-j2\pi tf/n) \]
What is the complexity of DFT?

\[ X_f = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t \exp(-j2\pi tf / n) \]

A: Naively, \( O(n^2) \)
However, if $n$ is a power of 2 (or a number with many divisors), we can make it $O(n \log n)$.

Main idea: if we know the DFT of the odd time-ticks, and of the even time-ticks, we can quickly compute the whole DFT.

Details: in Num. Recipes.
DFT - Conclusions

- It spots periodicities (with the ‘amplitude spectrum’)
- can be quickly computed ($O(n \log n)$), thanks to the FFT algorithm.
- standard tool in signal processing (speech, image etc signals)
WAVELET TRANSFORM

Based on material by Christos Faloutsos.
Reminder: Problem:

Goal: given a signal (eg., #packets over time)
Find: patterns, periodicities, and/or **compress**

![Graph](image)

- lynx caught per year
- (packets per day; virus infections per month)
Wavelets - DWT

- DFT is great - but, how about compressing a spike?
Wavelets - DWT

- DFT is great - but, how about compressing a spike?
- A: Terrible - all DFT coefficients needed!
Wavelets - DWT

- DFT is great - but, how about compressing a spike?
- A: Terrible - all DFT coefficients needed!
Wavelets - DWT

- Similarly, DFT suffers on short-duration waves (e.g., baritone, silence, soprano)
Wavelets - DWT

- Solution#1: Short window Fourier transform (SWFT)
- But: how short should be the window?
Wavelets - DWT

- Answer: **multiple** window sizes! -> DWT
Haar Wavelets

- subtract sum of left half from right half
- repeat recursively for quarters, eight-ths, ...
Wavelets - construction

\[ x_0 \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \]
Wavelets - construction

level 1  d1,0  

s1,0  d1,1  s1,1  .......

-  

x0  x1  x2  x3  x4  x5  x6  x7
Wavelets - construction

level 2  \( d_{2,0} \)

\[ \begin{align*}
    & s_{2,0} \\
    & \quad \downarrow \\
    & d_{1,0} \quad s_{1,0} \quad d_{1,1} \quad s_{1,1} \\
    & \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
    & x_0 \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7
\end{align*} \]
Wavelets - construction

\[ d_{2,0} \]

\[ s_{2,0} \]

\[ d_{1,0} \]

\[ s_{1,0} \]

\[ d_{1,1} \]

\[ s_{1,1} \]

\[ x_0 \]

\[ x_1 \]

\[ x_2 \]

\[ x_3 \]

\[ x_4 \]

\[ x_5 \]

\[ x_6 \]

\[ x_7 \]

etc ...

........
Wavelets - construction

Q: map each coefficient on the time-freq. plane

\[
\begin{align*}
    d_{2,0} & \quad s_{2,0} \\
    d_{1,0} & \quad s_{1,0} \quad d_{1,1} \quad s_{1,1} \\
    & \quad + \quad - \\
    & \quad x_0 \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7
\end{align*}
\]
Wavelets - construction

Q: map each coefficient on the time-freq. plane

\[ d_{2,0}, s_{2,0}, d_{1,0}, s_{1,0}, d_{1,1}, s_{1,1}, \ldots \]

\[ x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7 \]
Haar wavelets - code

#!/usr/bin/perl5
# expects a file with numbers
# and prints the dwt transform
# The number of time-ticks should be a power of 2
# USAGE
#    haar.pl <fname>

my @vals=();
my @smooth; # the smooth component of the signal
my @diff;  # the high-freq. component

# collect the values into the array @val
while(<>){
    @vals = ( @vals , split );
}

my $len = scalar(@vals);
my $half = int($len/2);
while($half >= 1 ){
    for(my $i=0; $i< $half; $i++){
        $diff [$i] = ($vals[2*$i] - $vals[2*$i + 1] )/ sqrt(2);
        print	', $diff[$i];
        $smooth [$i] = ($vals[2*$i] + $vals[2*$i + 1] )/ sqrt(2);
    }
    print
    @vals = @smooth;
    $half = int($half/2);
}
print	', $vals[0], 
;      # the final, smooth component
Observation 1:
‘+’ can be some weighted addition
‘-’ is the corresponding weighted difference
(‘Quadrature mirror filters’)  

Observation 2: unlike DFT/DCT,
there are *many* wavelet bases: Haar, Daubechies-4, Daubechies-6, Coifman, Morlet, Gabor, ...
Wavelets - how do they look like?

- E.g., Daubechies-4
Wavelets - how do they look like?

- E.g., Daubechies-4
Wavelets - how do they look like?

- E.g., Daubechies-4
Wavelets - Drill#1:

- Q: baritone/silence/soprano - DWT?
Wavelets - Drill#1:

- Q: baritone/silence/soprano - DWT?

![Wavelet Diagram]

value

time
Wavelets - Drill#2:

- Q: spike - DWT?
Wavelets - Drill#2:

- Q: spike - DWT?

```
0.00  0.00  0.71  0.00
0.00  0.50
-0.35
0.35
```

```
0.00    0.00
0.71
0.00
```
Wavelets - Drill#3:

- Q: weekly + daily periodicity, + spike - DWT?
Wavelets - Drill#3:

- Q: weekly + daily periodicity, + spike - DWT?
Wavelets - Drill#3:

- Q: weekly + **daily** periodicity, + spike - DWT?

![Wavelet Diagram](image)
Wavelets - Drill#3:

- Q: weekly + daily periodicity, + spike - DWT?
Wavelets - Drill#3:

- Q: weekly + daily periodicity, + spike - DWT?
Wavelets - Drill#3:

- Q: DFT?
Delta?

\[ x(0) = 1; \quad x(t) = 0 \text{ elsewhere} \]
Delta?

\[ x(0) = 1; \quad x(t) = 0 \text{ elsewhere} \]
2 cosines?

\[ x(t) = \cos\left( 2 \times \pi \times 4 \times t / 1024 \right) + 5 \times \cos\left( 2 \times \pi \times 8 \times t / 1024 \right) \]
2 cosines?

\[ x(t) = \cos\left( 2 \times \pi \times 4 \times t / 1024 \right) + 5 \cos\left( 2 \times \pi \times 8 \times t / 1024 \right) \]

Which one is for freq. = 4?
2 cosines?

\[ x(t) = \cos\left(2 \pi \frac{4 t}{1024}\right) + 5 \cos\left(2 \pi \frac{8 t}{1024}\right) \]

Which one is for freq. = 4?

f~8 → f
f~4 → t

Which one is for freq. = 4?
Chirp?

\[ x(t) = \cos\left( 2 \pi t^2 / 1024 \right) \]
Chirp?

\[ x(t) = \cos\left( 2 \pi t^2 / 1024 \right) \]
Chirp?

\[ x(t) = \cos\left(2 \pi t^2 / 1024\right) \]
Chirp?

\[ x(t) = \cos\left( 2 \times \pi \times t \times t / 1024 \right) \]
More examples (BGP updates)  
[Prakash+, 2009]
More examples (BGP updates)

Low freq.: omitted

freq.
More examples (BGP updates)

freq.
More examples (BGP updates)

Prolonged spike

freq.

Prakash 2015

CS 6604:DM Large Networks &
Time-Series
More examples (BGP updates)

15K msgs, for several hours: 6pm-4am

freq.
Wavelets - Drill

- Or use ‘R’, ‘octave’ or ‘matlab’ – R:

```R
install.packages("wavelets")
library("wavelets")
X1<-c(1,2,3,4,5,6,7,8)
dwt(X1, n.levels=3, filter="d4")
mra(X1, n.levels=3, filter="d4")
```
Wavelets - k-dimensions?

- easily defined for any dimensionality (like DFT, DCT)
Wavelets - example

Wavelets achieve *great* compression:

20 100 400 16,000
# coefficients
Wavelets - intuition

- Edges (horizontal; vertical; diagonal)
Wavelets - intuition

- Edges (horizontal; vertical; diagonal)
- recurse
Wavelets - intuition

- Edges (horizontal; vertical; diagonal)
Advantages of Wavelets

- Better compression (better RMSE with same number of coefficients)
- Closely related to the processing of the mammalian eye and ear
- Good for progressive transmission
- Handle spikes well
- Usually, fast to compute ($O(n)$!)
Overall Conclusions

- DFT spot periodicities
- DWT: multi-resolution - matches processing of mammalian ear/eye better
- Both: powerful tools for compression, pattern detection in real signals
- Both: included in math packages (matlab, R, mathematica, ... )