15-826: Multimedia Databases and Data Mining

Lecture #23: DSP tools – Fourier and Wavelets

C. Faloutsos

Must-read Material

- DFT/DCT: In *PTVF* ch. 12.1, 12.3, 12.4; in *Textbook* Appendix B.
- Wavelets: In *PTVF* ch. 13.10; in *MM Textbook* Appendix C

Outline

Goal: ‘Find similar / interesting things’
- Intro to DB
  - Indexing - similarity search
  - Data Mining
Indexing - Detailed outline

- primary key indexing
- ..
- multimedia
- Digital Signal Processing (DSP) tools
  - Discrete Fourier Transform (DFT)
  - Discrete Wavelet Transform (DWT)

DSP - Detailed outline

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

Introduction

Goal: given a signal (e.g., sales over time and/or space)
Find: patterns and/or compress

<table>
<thead>
<tr>
<th>count</th>
<th>lynx caught per year</th>
<th>year</th>
</tr>
</thead>
</table>
What does DFT do?
A: highlights the periodicities

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
  http://www.relisoft.com/freeware/freq.html
• 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

DFT: definition

- Discrete Fourier Transform (n-point):

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

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

How does it work?

Decomposes signal to a sum of sine (and cosine) waves.
Q: How to assess ‘similarity’ of \( x \) with a wave?

value

\[ x = \{x_0, x_1, ... x_{n-1}\} \]

time

0 1 n-1
How does it work?

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

value

freq. f=0

0 1 n-1 time

value

freq. f=1 (sin(t * 2*π/n) )

0 1 n-1 time

How does it work?

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

value

freq. f=2

0 1 n-1 time

How does it work?

‘basis’ functions

sine, freq =1

0 1 n-1

cosine, f=1

0 1 n-1

cosine, f=2

0 1 n-1
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: ~ all the similarities of x with the basis functions

Since \( e^{j\pi} = \cos(f) + j\sin(f) \)
\[ (j = \sqrt{-1}), \]
we finally have:

DFT: definition

- Discrete Fourier Transform (n-point):

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

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

inverse DFT

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

- **Good news:** Available in **all** symbolic math packages, e.g., in 'mathematica'
  
  ```math
  x = [1,2,1,2];
  X = Fourier[x];
  Plot[Abs[X];
  ```

DFT: definition

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

---

DFT: definition

(variations:

```math
\alpha_f = \sum_{i=0}^{N-1} x_i \cos(2\pi fi/N)
\beta_f = \sum_{i=0}^{N-1} x_i \sin(2\pi fi/N)
\begin{align*}
  f &= 0, 1, \ldots, N/2
\end{align*}
```
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:
\[ X_f = (X_{-f})^* \]

( “*”: 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 $|\phi_f| \text{ vs } f (f=0, 1, \ldots, n-1)$:

Anti-symmetric

(Rarely used)

DFT: examples

<table>
<thead>
<tr>
<th>time</th>
<th>freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>flat</td>
<td>Amplitude</td>
</tr>
</tbody>
</table>
DFT: examples

Low frequency sinusoid

---

DFT: examples

- Sinusoid - symmetry property: $X_f = X^*_{N-f}$

---

DFT: examples

- Higher freq. sinusoid
DFT: examples

$\text{DFT: examples}$

$\text{DFT: Amplitude spectrum}$

Amplitude: $A_j^2 = \text{Re}^2(X_j) + \text{Im}^2(X_j)$
DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?
DFT: Amplitude spectrum

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

- Let’s see it in action (defunct now…)
- plain sine
- phase shift
- two sine waves
- the ‘chirp’ function
- [http://ion.researchsystems.com/](http://ion.researchsystems.com/)

---

**Plain sine**

Number of samples: 256
Sampling rate: 8000 samples / s
Signal waveform expression: \( \sin(200\pi t) \)

![Graph of plain sine signal]

---

**Plain sine**

Number of samples: 256
Sampling rate: 8000 samples / s
Signal waveform expression: \( \sin(200\pi t) \)

![Graph of plain sine signal]
Two sines

Chirp
Chirp

Number of samples: 216
Sampling rate: 8000 samples/s
Sine wave frequency: \( \sin(2\pi f_0 t) \)

Another applet

http://www.falstad.com/fourier/
(seems virus-free – but scan, before you install)
FFT_applet/index.html

Properties

- Time shift sounds the same
  - Changes only phase, not amplitudes

- Sawtooth has almost all frequencies
  - With decreasing amplitude

- Spike has all frequencies
**DFT: Parseval’s theorem**

\[
s\sum x_i^2 = \sum |X_f|^2
\]

Ie., 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: \( x = \{ 0, 1, 0, 0 \} \ (n = 4) \)
- \( X_0 = ? \)
  - Value:
    - 1
    - 0
  - Time:
    - 0
    - 1

A:
- \( X_0 = \frac{1}{\sqrt{4}} \times 1 \times \exp(-j \frac{2 \pi 0}{n}) = \frac{1}{2} \)
- \( X_1 = ? \)
- \( X_2 = ? \)
- \( X_3 = ? \)

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)

Q: check Parseval's theorem
Arithmetic examples

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

Q: What does this mean?

- 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)

Observations

- DFT of 'step' function:
  \( x = \{ 0, 0, ..., 0, 1, 1, ... 1 \} \)

\[
\begin{align*}
  x_t & \quad \text{\ldots \ldots} \\
  f = 0 & \\
  t & 
\end{align*}
\]
Observations

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

\[ x_t \]
\[ f=1 \]
\[ f=0 \]
\[ t \]

• the more frequencies, the better the approx.
• ‘ringing’ becomes worse
• reason: discontinuities; trends

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

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

 original
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: \( X_3 = ? \)

check ‘symmetry’
check Parseval
Observations

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

Does this make sense?

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)
**DCT**

Discrete Cosine Transform

- motivation#1: DFT gives complex numbers
- motivation#2: how to avoid the ‘frequency leak’ of DFT on trends?

- brilliant solution to both problems: mirror the sequence, do DFT, and drop the redundant entries!

- (see Numerical Recipes for exact formulas)
DCT - properties

- it gives real numbers as the result
- it has no problems with trends
- it is very good when $x_i$ and $x_{i+1}$ are correlated

(Thus, is used in JPEG, for image compression)

DSP - Detailed outline

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

2-d DFT

- Definition:

$$X_{f_1, f_2} = \frac{1}{\sqrt{n_1}} \frac{1}{\sqrt{n_2}} \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} x_{i,j} \exp(-2\pi j f_1 n_1) \exp(-2\pi j f_2 n_2)$$
2-d DFT
- Intuition:
  - do 1-d DFT on each row
  - and then 1-d DFT on each column

2-d DFT
- Quiz: how do the basis functions look like?
  - for f1 = f2 = 0
  - for f1 = 1, f2 = 0
  - for f1 = 1, f2 = 1

2-d DFT
- Quiz: how do the basis functions look like?
  - for f1 = f2 = 0  flat
  - for f1 = 1, f2 = 0  wave on x; flat on y
  - for f1 = 1, f2 = 1  ~ egg-carton
2-d DFT

• Quiz: how do the basis functions look like?
• for $f_1 = f_2 = 0$ flat
• for $f_1 = 1, f_2 = 0$ wave on $x$; flat on $y$
• for $f_1 = 1, f_2 = 1$ ~ egg-carton

DSP - Detailed outline

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

FFT

• What is the complexity of DFT?

$$X_j = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} x_i \exp\left(-j2\pi tf/n \right)$$
What is the complexity of DFT?

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

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)
Detailed outline

- primary key indexing
- ...
- multimedia
- Digital Signal Processing (DSP) tools
  - Discrete Fourier Transform (DFT)
  - Discrete Wavelet Transform (DWT)

Reminder: Problem:

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

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!

![Graph showing value over time and frequency](image)

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

\[ x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \]

Level 1:
- \( d_{1,0} \)
- \( s_{1,0} \)
- \( d_{1,1} \)
- \( s_{1,1} \)

\[ ... \]

Level 2:
- \( d_{2,0} \)
- \( s_{2,0} \)
- \( d_{1,0} \)
- \( s_{1,0} \)
- \( d_{1,1} \)
- \( s_{1,1} \)

\[ ... \]
Wavelets - construction

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

\[ d_{2,0} \quad s_{2,0} \]
\[ d_{1,0} \quad s_{1,0} \quad d_{1,1} \quad s_{1,1} \]
\[ x_0 \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \]
Haar wavelets - code

```perl
#!/usr/bin/perl

# expects a file with numbers
# and prints the dwt transform
# The number of time-ticks should be a power of 2

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],
```

Wavelets - construction

Observation1:
- '+' can be some weighted addition
- '-' is the corresponding weighted difference ('Quadrature mirror filters')

Observation2: 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 - Drill#1:

- Q: baritone/silence/soprano - DWT?

- Diagram of wavelet coefficients in time and frequency domains.
Wavelets - Drill#1:
- Q: baritone/silence/soprano - DWT?

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
Wavelets - Drill#3:

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

\[ f \]
\[ t \]
Wavelets - Drill#3:
• Q: weekly + daily periodicity, + spike - DWT?
Wavelets - Drill:

Let’s see it live:

http://dsp.rice.edu/software/dsp-teaching-tools
delta; cosine; cosine2; chirp
- Haar vs Daubechies-4, -6, etc

Delta?

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

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

Which one is for freq.=4?
Chirp?

\[ x(t) = \cos(2\pi \frac{t^2}{1024}) \]
Chirp?

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

More examples (BGP updates)

BGP-lens: Patterns and Anomalies in Internet Routing Updates
B. Aditya Prakash et al, SIGKDD 2009

More examples (BGP updates)

Low freq.: omitted

freq.
More examples (BGP updates)

freq.

Prolonged spike

freq.

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

[Link](http://grail.cs.washington.edu/projects/query/)

Wavelets achieve “great” compression:
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, DCT spot periodicities
- DWT: multi-resolution - matches processing of mammalian ear/eye better
- All three: powerful tools for compression, pattern detection in real signals
- All three: included in math packages (matlab, R, mathematica, ...)

**Resources**

- Numerical Recipes in C: great description, intuition and code for all three tools
- *xwpl*: open source wavelet package from Yale, with excellent GUI.
Resources (cont’d)

• (defunct?)
  http://www.dsptutor.freeuk.com/jsanalyser/FFTSpectrumAnalyser.html : Nice java applets

• http://www.relisoft.com/freeware/freq.html : voice frequency analyzer (needs microphone – MSwindows only)

Resources (cont’d)

• www-dsp.rice.edu/software/EDU/mra.shtml (wavelets and other demos)
• R (*install.packages("wavelets") )