15-826: Multimedia Databases and Data Mining

DSP tools: Fourier and Wavelets
C. Faloutsos

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 (eg., sales over time and/or space)
Find: patterns and/or compress

What does DFT do?

A: highlights the periodicities
Why should we care?

A: several real sequences are periodic
Q: Such as?

- sales patterns follow seasons;
- economy follows 50-year cycle
- temperature follows daily and yearly cycles
Many real signals follow (multiple) cycles

For example: human voice!

- Frequency analyzer
- speaker identification
- impulses/noise -> flat spectrum
- high pitch -> high frequency

‘Frequency Analyzer’

DFT and stocks

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

Log(ampl)
**DFT: definition**

- Discrete Fourier Transform (n-point):
  \[ X_j = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t \exp(-j 2\pi t f / n) \]
  \( (j = \sqrt{-1}) \)
- Inverse DFT:
  \[ x_t = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} X_j \exp(+j 2\pi t f / n) \]

**How does it work?**

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

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

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 (~cosine similarity)

**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
How does it work?

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

DFT: definition

- **Good news:** Available in all symbolic math packages, e.g., in ‘mathematica’
  \( x = \{1, 2, 1, 2\}; \)
  \( X = \text{Fourier}[x]; \)
  \( \text{Plot[Abs}[X] ]; \)

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

- **Discrete Fourier Transform (n-point):**
  \[
  X_f = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} x_i \cdot \exp(-j2\pi tf/n)
  \]
  \((j=\sqrt{-1})\)

  - **Inverse DFT**
    \[
    x_i = \frac{1}{\sqrt{n}} \sum_{f=0}^{n-1} X_f \cdot \exp(+j2\pi tf/n)
    \]

DFT: definition

- **Observation - SYMMETRY property:**
  \[
  X_f = (X_{n-f})^* \\
  \text{(*: complex conjugate: } (a + b j)^* = a - b j \text{)}
  \]
**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: examples**

**DFT: Amplitude spectrum**

Amplitude: \( A_f^2 = \text{Re}^2(X_f) + \text{Im}^2(X_f) \)

**DFT: definition**

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

Anti-symmetric

(Rarely used)
DFT: examples

- Sinusoid - symmetry property: $X_j = X^*_{n-j}$

DFT: examples

- Higher freq. sinusoid

DFT: examples

- Intuition: strength of frequency $f$

DFT: Amplitude spectrum

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

DFT: Amplitude spectrum

- Amplitude $A_j^2 = \text{Re}(X_j) + \text{Im}(X_j)$
- Intuition: strength of frequency $f$
**DFT: Amplitude spectrum**

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

• 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

• Let's see it in action!
  • http://www.dsptutor.freeuk.com/jsanalyser/FTFSpectrumAnalyser.html
  • plain sine
  • phase shift
  • two sine waves
  • the 'chirp' function

DFT: Parseval’s theorem

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

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

X
\[ x = \{ x0, x1 \} \]

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
\[ \begin{array}{c}
  \text{time} \\
  0 \quad 1
\end{array} \]

1

Arithmetic examples

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

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

Arithmetic examples

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

Arithmetic examples

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

Arithmetic examples

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

Arithmetic examples

- Q: What does this mean?

Arithmetic examples

- Q: does the ‘symmetry’ property hold?
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)

Observations

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

Observations

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

Observations

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

Observations

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

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

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

\[ x_t \]

\[ t \]

\[ \text{Observations} \]

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

\[ x_t \]

\[ t \]

\[ \text{Observations} \]

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

\[ x_t \]

\[ t \]

\[ \text{Observations} \]

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

\[ \text{Observations} \]

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

\[ 0 \quad 1 \quad 2 \quad f \]

\[ A_f \]

• 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!

DCT - properties

- it gives real numbers as the result
- it has no problems with trends
- it is very good when \( x_t \) and \( x_{(t+1)} \) 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:

\[ X_{a,b} = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} \exp \left( -j \frac{2\pi}{M} am \right) \exp \left( -j \frac{2\pi}{N} bn \right) \]

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 \sim 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?

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

- A: Naively, O(n^2)

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

<table>
<thead>
<tr>
<th>count</th>
<th>lynx caught per year (packets per day; virus infections per month)</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 (eg., 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

$$x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7$$

Wavelets - construction

level 1

$$d_{1,0} \quad s_{1,0} \quad d_{1,1} \quad s_{1,1} \quad \ldots$$

$$x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7$$

Wavelets - construction

level 2

$$d_{2,0} \quad s_{2,0}$$

$$d_{1,0} \quad s_{1,0} \quad d_{1,1} \quad s_{1,1} \quad \ldots$$

$$x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7$$

Wavelets - construction

etc ...

$$d_{2,0} \quad s_{2,0}$$

$$d_{1,0} \quad s_{1,0} \quad d_{1,1} \quad s_{1,1} \quad \ldots$$

$$x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7$$

Wavelets - construction

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

f

t
Wavelets - construction

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

f

d2.0
s2.0

d1.0
s1.0
s1.1

x0 x1 x2 x3 x4 x5 x6 x7

Wavelets - construction

Observation1:
- ‘d’ can be some weighted addition
- ‘s’ 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, ...

Haar wavelets - code

```perl
my @diff;   # the high-freq. component
my @smooth; # the smooth component of the signal

# collect the values into the array @val
my $half = int($len/2);

@vals = ( @vals , split );
print "\n";
$diff[$i] = ($vals[2*$i] - $vals[2*$i + 1] )/ sqrt( 2);
```

Wavelets - how do they look like?

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

- Q: baritone/silence/soprano - DWT?

Wavelets - Drill#2:

- Q: spike - DWT?

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?

Wavelets - Drill#3:

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

Wavelets - Drill#3:

- Q: DFT?

Wavelets - k-dimensions?

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

Wavelets - Drill:

Let’s see it live:

http://monet.me.ic.ac.uk/people/gavin/java/waveletDemos.html

- delta; cosine; cosine2; chirp
- Haar vs Daubechies-4, -6, etc
- or
- http://www-dsp.rice.edu/~harry/class/wavelet/wavejava.html
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, ...)

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)

- http://www.relisoft.com/freeware/ freq.html: Voice frequency analyzer (needs microphone)

Resources (cont’d)

- http://monet.me.ic.ac.uk/people/gavin/java/waveletDemos.html: Wavelets and scalograms