Wavelet Synopses

Kyueseok Shim

Seoul National University

http://ee.snu.ac.kr/~shim

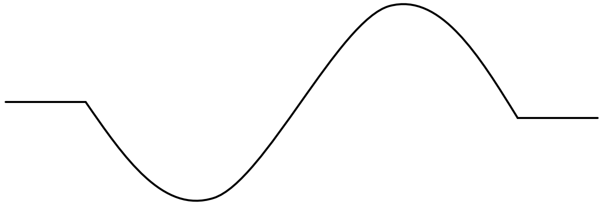
The Synopsis Construction Problem

- Formally, given a signal X and a dictionary $\{\psi_i\}$ find a representation $F=\sum_i z_i \psi_i$ with at most B non-zero z_i minimizing some error which a fn of X-F
- In case of histograms the “dictionary“ was the set of all possible intervals – but we could only choose a non-overlapping set.

The eternal “what if”

- If the $\{\psi_i\}$ are “designed for the data” do we get a better synopsis ?

- Absolutely!
- Consider a Sine wave …
- Or any smooth fn.



- Why though ?

Representations not piecewise const.

- Electromagnetic signals are sine/cosine waves.
- If we are considering any process which involve electromagnetic signals - this is a great idea.
- These are particularly great for representing periodic functions.
- Often these algorithms are found in DSP (digital signal processing chips)
- A fascinating 300+ years of history in Math !

A slight problem …

- νι νιll cφmε βαcκ τφ Fφυrιεr

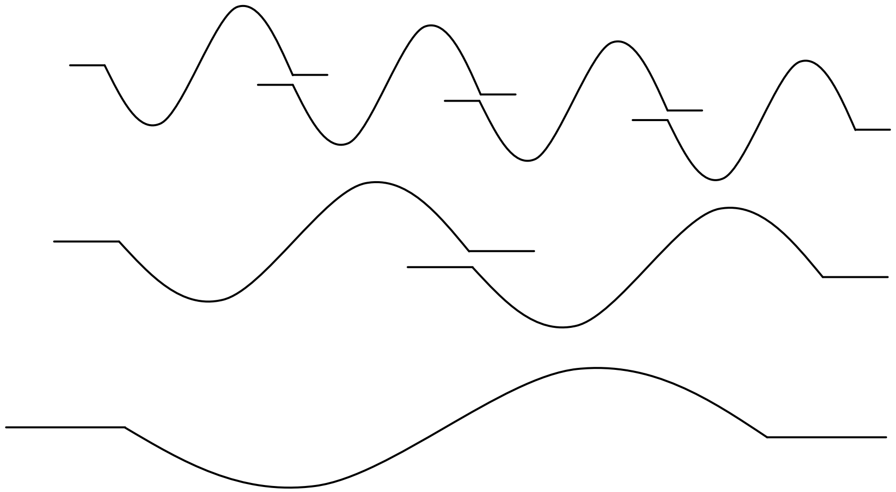
- Fourier is suitable to smooth “natural processes“

- If we are talking about signals from man-made processes, clearly they cannot be natural (and hardly likely to be smooth) …

- More seriously, discreteness and burstiness…

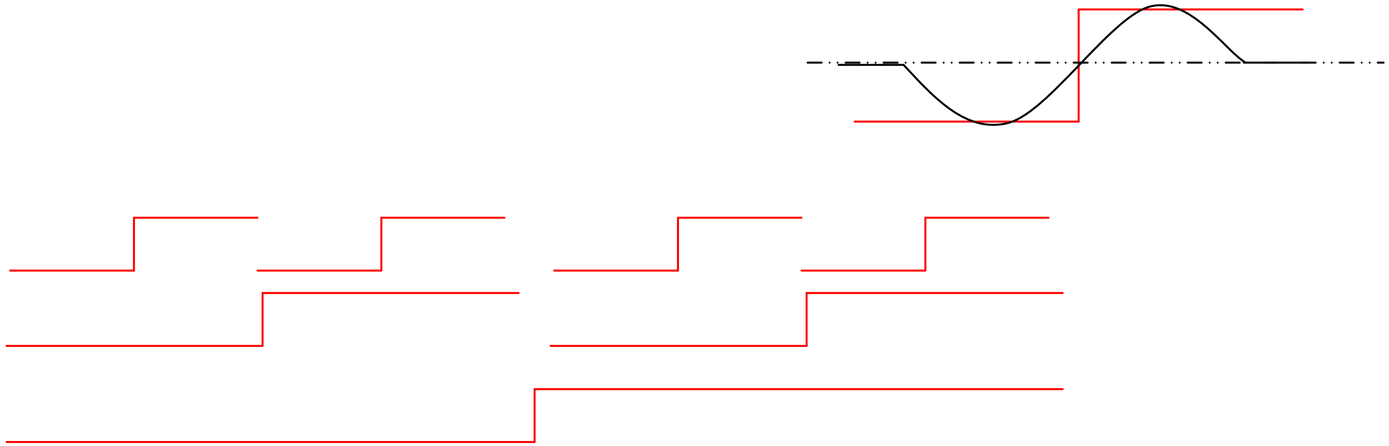
The Wavelet (frames)

- Inherits properties from both worlds
- Fourier transform has all frequencies.
- Considers frequencies that are powers of 2 but the effect of each wave is limited (shifted)



Wavelets

- What to do in a discrete world ?



The Haar Wavelets (1910) !

The Haar Wavelets

- Best “energy“ synopsis amongst all wavelets
- Great for data with discontinuities.
- A natural extension to discrete spaces
  - {1,-1,0,0,0,0…}, {0,0,1,-1,0,0,…},{0,0,0,0,1,-1,…}…
  - {1,1,-1,-1,0,0,0,0,…},{0,0,0,0,1,1,-1,-1,…}…

Wavelet

- A useful mathematical tool for hierarchically decomposing functions
- Represent a function in terms of
  - A coarse overall shape
  - Details that range from broad to narrow
- Haar wavelet
  - The Haar basis is the simplest wavelet basis
  - Fastest to compute and easiest to implement

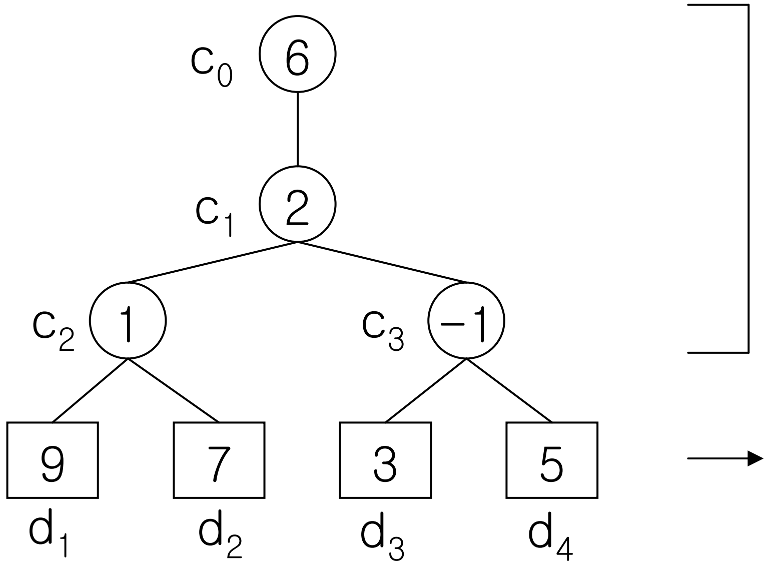
Haar wavelet

- Given a one dimensional data with a resolution 4, [9 7 3 5]
  - Recursive pairwise averaging and differencing at different resolutions

Resolution	Averages	Detail coefficients
4	[9 7 3 5]	
2	[8 4]	[1 −1]
1	[6]	[2]

  - The wavelet transform of the original data is given by [6 2 1 -1]

Error Tree



Wavelet Coefficients

Original data values

Path(u) : The set of all nodes in T that are proper ancestor of u with nonzero coefficients (definition in [GG02])

Reconstruction

- The reconstruction of any data value d_i using Error Tree

$$d_i = \sum_{c_j \in path(d_i)} \delta_{ij} \cdot c_j$$

  - Where $\delta_{ij} = +1$ if $d_i \in$ left leaves of c_j , or j=0, and $\delta_{ij} = -1$ otherwise

Ex) in previous error tree

$d_3 = c_0 - c_1 + c_3 = 6 - 2 + (-1) = 3$

Compression

- Wavelet compression
  - A large number of the detail coefficients turn out to be very small in magnitude
  - Removing these small coefficients introduces small errors
  - Lossy compression

Ex) from [6 2 1 -1], take two coefficients, 6, 2, that is [6 2 0 0] then

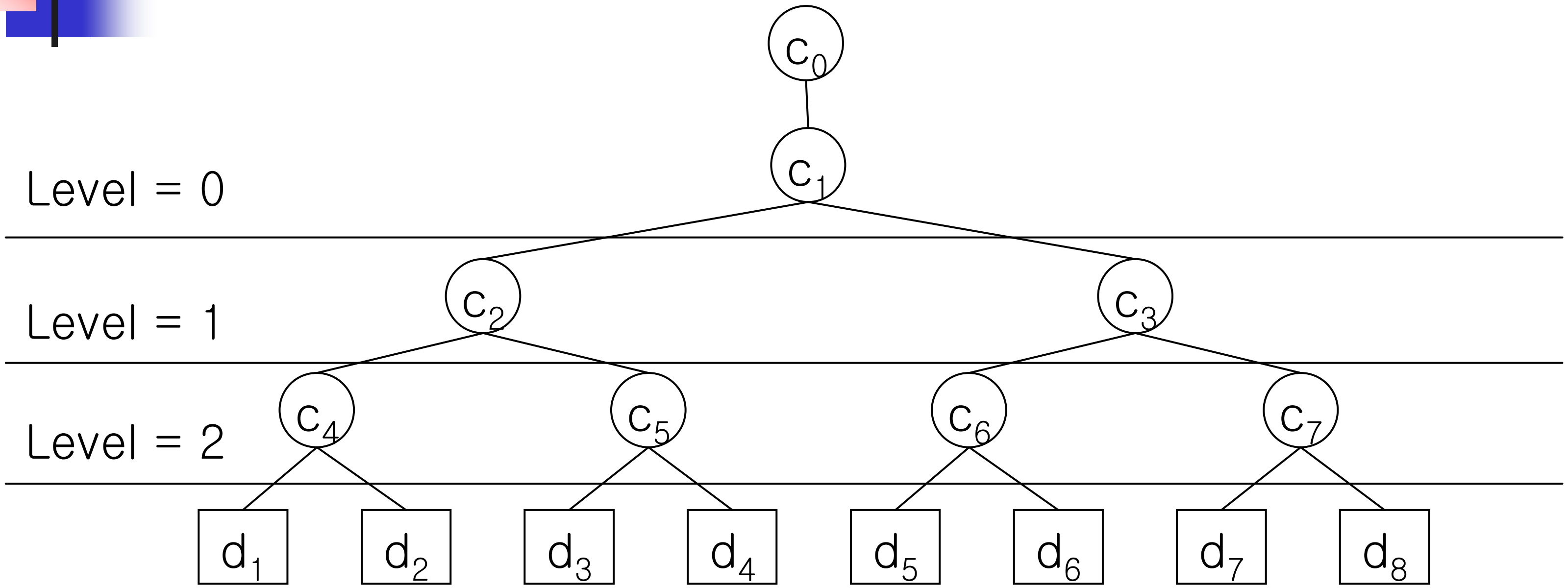
original data = [9 7 3 5]

reconstructed data = [8 8 4 4]

Normalization

- In order to equalize the importance of all wavelet coefficients
  - Normalizing the coefficients is needed
- If the coefficients have the same importance
  - We could choose the coefficients in order of absolute magnitude
  - Then we could achieve the best approximation of original data

Example error tree



Remove					L² error
c_2	$-c_2$	$-c_2$	c_2	c_2	$4*c_2^2$
c_5			$-c_5$	c_5	$2*c_5^2$
c_2,c_5	$-c_2$	$-c_2$	c_2-c_5	c_2+c_5	$4*c_2^2+2*c_5^2$

Haar wavelet normalization

- Assume we use L^2 error
  - If we remove c_2, then it affects four values d_1, d_2, d_3, d_4 and results in $4*c_2^2$ L^2 error
  - If we remove c_5, then it affects two values d_5, d_6 and result in $2*c_5^2$ L^2 error
  - If we remove c_2, c_5, then it result in $4*c_2^2 + 2*c_5^2$ L^2 error
    - Removing each coefficient affects the L^2 error independently

Haar wavelet normalization(cont'd)

- If the values of c_2, c_5 are the same in absolute magnitude
  - Removing c_2 increases L^2 error more than removing c_5
- To compare the importance between c_2 and c_5 directly
  - we need to normalize the coefficients
  - If $4*c_2^2 = 2*c_5^2$, $c_2 = \frac{1}{\sqrt{2}} c_5$
  - c_2 and $\frac{1}{\sqrt{2}} c_5$ have the same importance

Haar wavelet normalization(cont’d)

- The coefficients in the same level have the same importance
- Between two coefficients which has one level difference
  - The higher level coefficients have $\frac{1}{\sqrt{2}}$ times importance of the lower level
- To normalize coefficients
  - Divide each wavelet coefficient by $\sqrt{2^l}$, where l denotes the level
  - Ex) [6 2 1 -1] $\xrightarrow{\text{Normalization}}$ [6 2 $\frac{1}{\sqrt{2}}$ $-\frac{1}{\sqrt{2}}$]

Minimize L^2 error in Haar wavelet compression

- Compressing the original N data using B(<<N) wavelet coefficients
  - Normalize the coefficients
  - Choose the B wavelet coefficients with the largest absolute value
  - This is an optimal method of minimizing L^2 error using B wavelet coefficients

Wavelets (2-D Harr Wavelets)

- Standard decomposition
  - Apply 1-D wavelet transformation to each row
  - Apply 1-D wavelet transformation to each column
- Non-standard decomposition
  - apply one step of 1-D wavelet transformation to each row and column repeatedly

Example (2-D Harr Wavelet)

$$\begin{array}{|cccc|} \hline 1.0 & 2.0 & 1.0 & 2.0 \\ 3.0 & 4.0 & 3.0 & 4.0 \\ 1.0 & 2.0 & 1.0 & 2.0 \\ 3.0 & 4.0 & 3.0 & 4.0 \\ \hline \end{array} \longrightarrow \begin{array}{|cccc|} \hline 2.5 & 0.5 & 2.5 & 0.5 \\ 1.0 & 0.0 & 1.0 & 0.5 \\ 2.5 & 0.5 & 2.5 & 0.5 \\ 1.0 & 0.0 & 1.0 & 0.5 \\ \hline \end{array}$$

$$\begin{array}{|cc:c|} \hline 1.0 & 2.0 & \\ 3.0 & 4.0 & \\ \hdashline & & \\ & & \\ \hline \end{array} \longrightarrow \begin{array}{|cc:c|} \hline 1.5 & 0.5 & \\ 3.5 & 0.5 & \\ \hdashline & & \\ & & \\ \hline \end{array} \longrightarrow \begin{array}{|cc:c|} \hline 2.5 & 0.5 & \\ 1.0 & 0.0 & \\ \hdashline & & \\ & & \\ \hline \end{array}$$

Example (2-D Harr Wavelet)

$$\boxed{\begin{matrix} 2.5 & 0.5 & 2.5 & 0.5 \\ 1.0 & 0.0 & 1.0 & 0.5 \\ 2.5 & 0.5 & 2.5 & 0.5 \\ 1.0 & 0.0 & 1.0 & 0.5 \end{matrix}} \longrightarrow \boxed{\begin{matrix} 2.5 & 2.5 & 0.5 & 0.5 \\ 2.5 & 2.5 & 0.5 & 0.5 \\ 1.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 1.0 & 0.0 & 0.0 \end{matrix}}$$

The Haar Synopsis Problem

- Formally, given a signal X and the Haar basis $\{\psi_i\}$ find a representation $F=\sum_i c_i \psi_i$ with at most B non-zero c_i minimizing some error which a fn of $X-F$

- Lets begin with the VOPT error $(||X-F||_2^2)$

The Magic of Parseval (no spears)

- The l_2 distance is unchanged by a rotation.
- A set of basis vectors $\{\psi_i\}$ define a rotation iff
  - h ψ_i,ψ_j i = δ_{ij} , i.e., $\begin{cases} 0 & \text{If } i \neq j \\ 1 & \text{If } i = j \end{cases}$

- Redefine the basis (scale) s.t. $||\psi_i||_2 = 1$
- Let the transform be W
- Then **$||X\text{-}F||_2 = ||\ W(X\text{-}F)||_2 = ||W(X) - W(F)||_2$**

- Now $W(F)=\{z_1,z_2,\ldots z_n\}$ and so
- $||W(X) - W(F)||_2 = \Sigma_i\ (W(X)_i - z_i)^2$

What did we achieve ?

- Storing the largest coefficients is the best solution.
- Note that the fact $z_i=W(X)_i$ is a consequence of the optimization and IS NOT a specification of the problem.
- More on that later.

Similar Time Sequences

- Given:
  - A set of time-series sequences
- Find
  - All sequences similar to the query sequence
  - All pairs of similar sequences

    whole matching vs. subsequence matching
- Sample Applications
  - Financial market
  - Market basket data analysis
  - Scientific databases
  - Medical Diagnosis

Whole Sequence Matching

Basic Idea

- Extract k features from every sequence
- Every sequence is then represented as a point in k-dimensional space
- Use a multi-dimensional index to store and search these points
    - Spatial indices do not work well for high dimensional data

      (i.e. Dimensionality curse:

      [Hellerstein, Koutsoupias, Papadimitrou 98])

Dimensionality Curse

Distance-Preserving Orthonormal Transformations

- Data-dependent
  - Need all the data to determine transformation
  - Example: K-L transform, SVD transform
- Data-independent
  - The transformation matrix is determined apriori
  - Example: DFT, DCT, Haar wavelet transform
  - DFT does a good job of concentrating energy in the first few coefficients

Why work with a few coefficients?

- If we keep only first a few coefficients in DFT, we can compute the lower bounds of the actual distance.

By Parseval's Theorem

The distance between two signals in the time domain is the same as their euclidean distance in the frequency domain.

- However, we need post-processing to compute actual distance and discard false matches.

Similar Time Sequences

- [Agrawal, Faloutsos, Swami 93]
- Take Euclidean distance as the similarity measure
- Obtain Discrete Fourier Transform (DFT) coefficients of each sequence in the database
- Build a multi-dimensional index using first a few ε Fourier coefficients
- Use the index to retrieve sequences that are at most distance away from query sequence
- Post-processing:
  - compute the actual distance between sequences in the time domain

References

- H. V. Jagadish, Nick Koudas, S. Muthukrishnan, Viswanath Poosala, Kenneth C. Sevcik, Torsten Suel: Optimal Histograms with Quality Guarantees. VLDB 1998: 275-286
- Yossi Matias, Jeffrey Scott Vitter, Min Wang: Wavelet-Based Histograms for Selectivity Estimation. SIGMOD Conference 1998: 448-459
- Sudipto Guha, Nick Koudas, Kyuseok Shim: Data-streams and histograms. STOC 2001: 471-475
- Minos N. Garofalakis, Phillip B. Gibbons: Wavelet synopses with error guarantees. SIGMOD Conference 2002: 476-487