An Algorithmer's View of Sparse Approximation Problems

S. *Muthu* Muthukrishnan

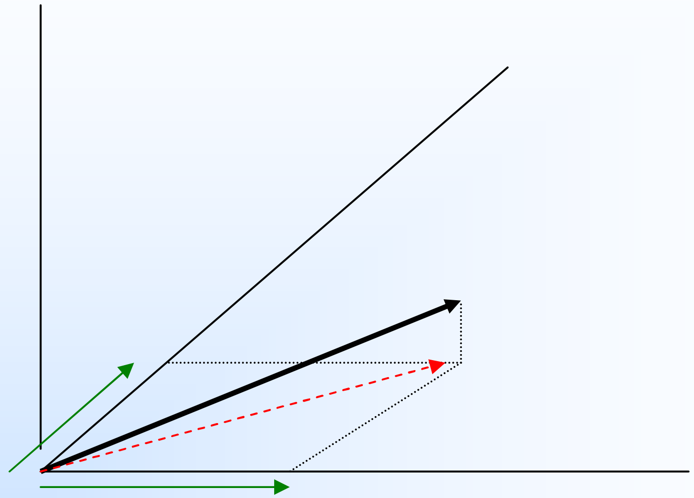
Google

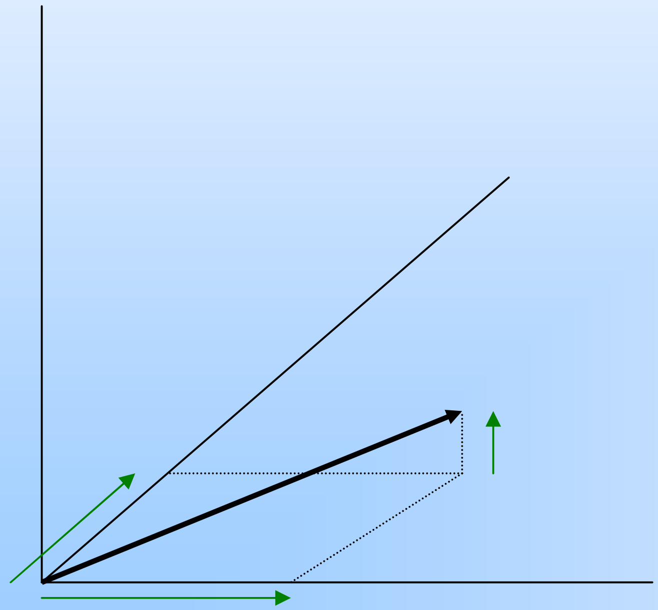
adorisms, mysliceofpizza

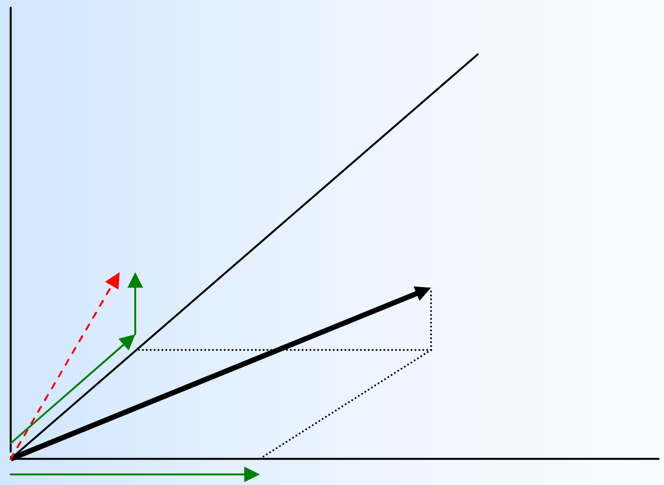
Talk Overview

- The Sparse Approximation Problem
  - Math vs Algorithms.
- Two Classical Extremes
  - Parseval world: orthonormal dictionaries.
  - Cook-Karp-Levin world: general dictionaries.
- Modern Versions
  - Compressed sensing.
  - Non uniform sparse approximation theory.
  - Incoherent dictionaries.
- Open Problems

Sparse Approximation Problem: Toy Example







Sparse Approximation Problem

- Given dictionary D of N dimensional vectors that span $\Re^N$. May be preprocessed, may have special properties.

- Query is a vector called the signal **A**[1,…,N].

- Sparse representation for **A** using b vectors of D:

$$R_b = \sum_{d_i \in D,\, i \in \Lambda,\, |\Lambda| \le b} \alpha_i d_i$$

- Minimize

$$\| \mathrm{A} - R_b \|_2^2 = \sum_j (\mathrm{A}[j] - \overline{\mathrm{A}[j]})^2$$

Sparse Approximation: Algorithms vs Math.

Algorithms:

- Given arbitrary signal $\mathbf{A}$, design an algorithm that finds a representation with error $\|\mathbf{A}-\mathbf{R}_b\|$ that approximates

$$\| A - R_b^* \|$$

Applied Mathematics:

- Given signal $\mathbf{A}$, that is p-compressible, that is,

  $\|\mathbf{A}-\mathbf{R}^*_b\| = O(b^{1-2/p})$

  determine an algorithm that computes a representation $\mathbf{R}$ that approximates $\mathbf{A}$ to error $\|\mathbf{A}-\mathbf{R}\| = O(b^{1-2/p})$
- Art of finding suitable D for an application.

Two Extremes

- World of Parseval
- World of Cook-Levin-Karp.

Parseval's World

- Dictionary is an orthonormal basis for $\Re^N$, ie N unit vectors ψ_i so $<\psi_i, \psi_j> = 1$ iff i=j, 0 otherwise.
- We have $\mathbf{A} = \sum_j \langle \mathbf{A}, \psi_j \rangle \psi_j \qquad c_j = \langle \mathbf{A}, \psi_j \rangle$
- Best b coeff of smallest error:

$$\mathbf{R}_b = \sum_{j \in \Lambda; |\Lambda| = b} c_j \; \psi_j \qquad \min \quad \| \mathbf{A} - \mathbf{R}_b \|$$

  - Since (Parseval's)

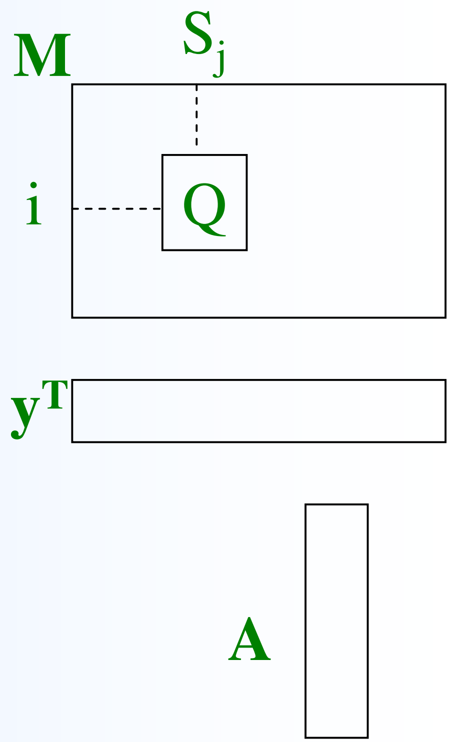
$$\| \mathbf{A} - \mathbf{R} \| = \sum_{i \in \Lambda} (c_i - \langle \mathbf{A}, \psi_i \rangle)^2 + \sum_{i \notin \Lambda} \langle \mathbf{A}, \psi_i \rangle^2$$

pick b largest $|\langle A, \psi_i \rangle|$

- Computationally easy.

Cook-Karp-Levin

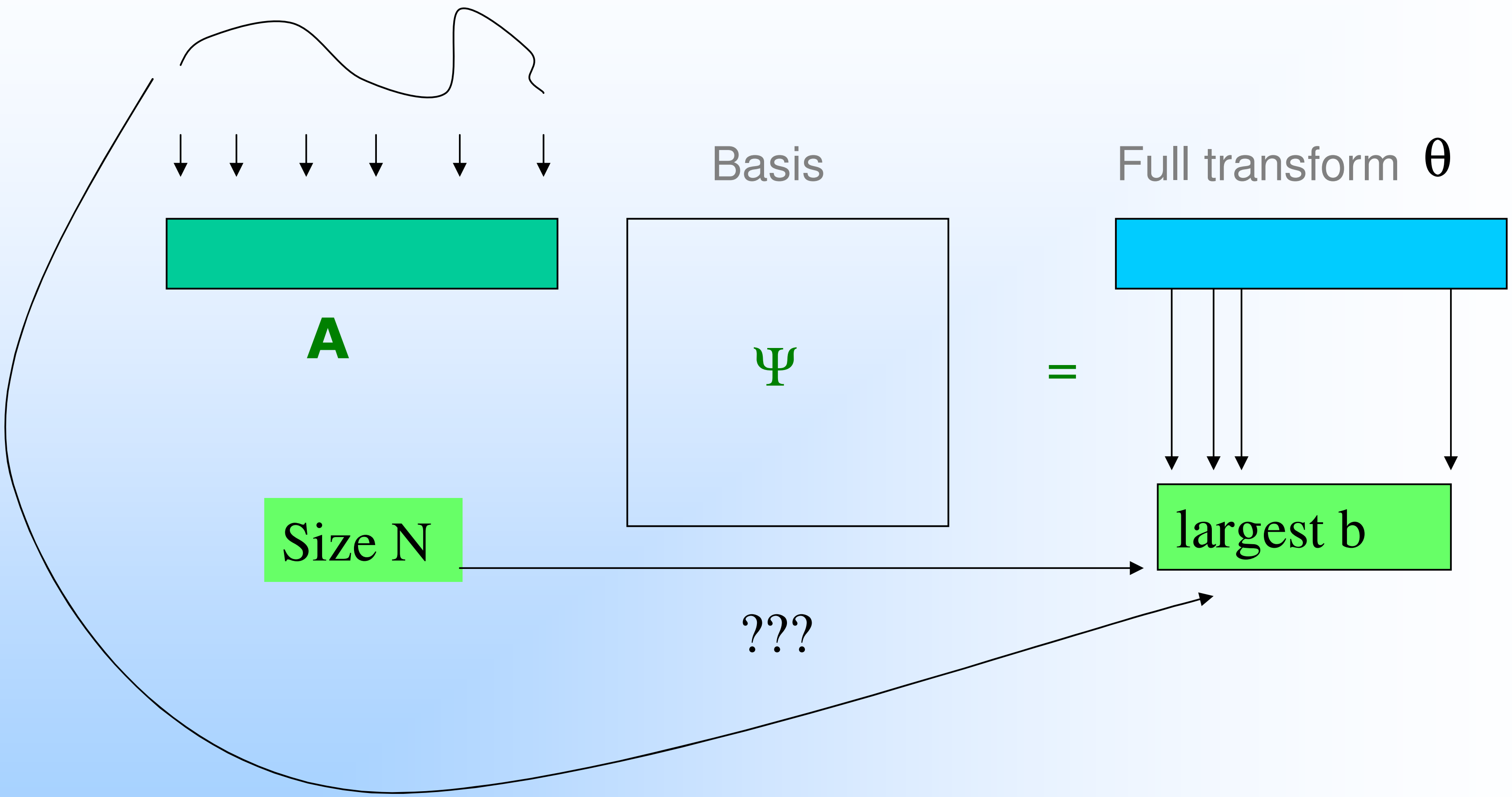
- Arbitrary dictionary **D.**
- Reduction from SAT to set cover such that:
  - YES -> there is an exact set cover of size ηd. ($\eta<1$).
  - NO -> no set cover of size d.
- Given elements [1,n] and sets $S_1, S_2, .. S_m$, we form **M**[i,j]=Q if i is in S_j. Let **A** be all 1's vector.
  - YES -> **y**[i]=1 if S_i is in exact set cover. **My**-**A**=0.
  - NO -> for any d vector solution to the sparse approx problem, there exists an i such that **My**[i]-**A**[i]=-Q.
- NP hard to solve it exactly. Hard to get (log N, L) approximation for any L. [N95, DM97, Mu05]



Modern Versions of Sparse Approximation Problem

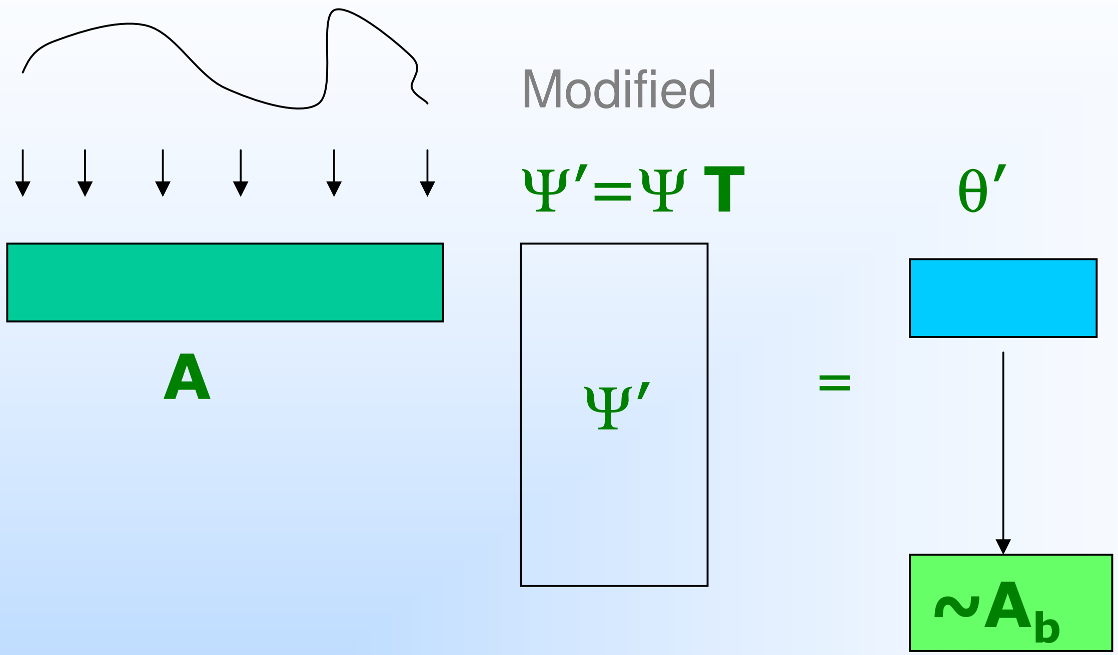
- Compressed sensing
- Weighted and other norms
- Incoherent dictionaries

Compressed Sensing, Insight



Donoho 04: What is the smallest number of measurements (inner products) needed?

Compressed Sensing



- Are there suitably small **T**? f(b,N)
- Can we construct such **T**?
- Can we decode from θ’ to $\sim A_b$ fast?

Compressed Sensing: Known Results

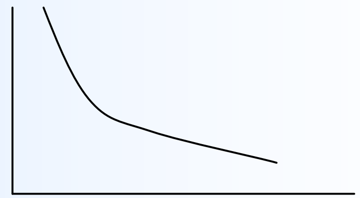
- There exists an N x $O(b \log N)$ matrix **T**.

Nonadaptive.
Random **T** will do.

- Consider any p-compressible signal **A**, p in (0,1).

$\|\mathbf{A}\text{-}\mathbf{R}^*_{\mathbf{b}}\| = O(b^{1-2/p})$

$|\theta_i|$



Sorted order of $|\theta_i|$'s

- From **AT**, solve the LP

$$\min_{g \in \Re^N} \|\mathbf{R}\|_1$$

$$\mathbf{RT} = \mathbf{AT}$$

Time poly in N, b.

- Claim:

$$\|\mathbf{A}\text{-}\mathbf{R}\| = O(b^{1-2/p})$$

O(1) approx
in the "worst case".

Donoho, Candes, Tao, Romberg, Vershynin,…. 2004+

Compressed Sensing: Excitement

- Excitement: “surprising”, “amazing”,…
- Math: **T** needs linear algebraic, geometric or uniform uncertainty principles.
- Applications: MR imaging, wireless communication,
- Connections and Extensions: Error correction, noisy measurements, distributed setting, etc.
- Thanks to Ron DeVore and Ingrid Daubauchies for simplifying and explaining things to me.

Compressed Sensing: An Algorithmer's View

- There exists an $N \times O(b \log N)$ matrix **T**.
- Consider any p-compressible signal **A**.
- From **AT**, solve the LP

$$\min_{g \in \Re^N} \|\mathbf{R}\|_1$$

$$\mathbf{RT} = \mathbf{AT}$$

- Claim: $\|\mathbf{A}-\mathbf{R}\| = O(b^{1-2/p})$

Why inner products?
How to construct **T**?

Why not any signal?

Different Algorithm?
Faster running time?

Why not wrt $\|\mathbf{A}-\mathbf{R}_b^*\|$?
What is **R** like?

Using random projections to retrieve largest **A**[i]'s: lot of prior work in group testing, learning theory, streaming algorithms.

Compressed Sensing: Our Results [Cormode, M 05]

- We can
  - in time poly(N, b, 1/ε), construct
  - a poly(b, log N, 1/ε) sized **T.** ← blog N
  - Given any p-compressible signal **A**,
  - in time poly(b, log N, 1/ε); ← poly(N)
  - we can construct $\mathbf{R}_b$ such that

$$\| \mathbf{A} - \mathbf{R}_b \| \leq \| \mathbf{A} - \mathbf{R}_b^* \| + \varepsilon b^{(1-2/p)}$$ ← $O(b^{(1-2/p)})$

First polynomial time construction known.

Compressed Sensing: Our Results [Cormode, M 05]

- Given any signal **A**,
  - We can construct a $N \times O(b/\varepsilon^2 \log^2 N \log(1/\delta))$ sized random **T**
  - Given **AT**, we can construct $\mathbf{R}_b$ such that with probability at least $1\text{-}\delta$,

$$\| \mathbf{A} - \mathbf{R}_b \| \leq (1+\varepsilon) \| \mathbf{A} - \mathbf{R}_b^* \|$$

Smallest known size of **T** thus far in the legacy of results in group testing, learning theory and streaming algorithms.

Main idea of our algorithm

- Each inner product gives the sum of a group of coefficients. Goal is to find approximately the b largest coefficients (in magnitude).
- Round 1: Group coefficients so that any set of poly(b) coefficients are separated.
  - Identify the coefficient that has the majority magnitude in each group via log N inner products. The identified set is a superset of the b largest coeff.
  - Key to the proof: the remainder coefficients together have small sum, so their combined effect is negligible.
- Round 2: Group coefficients so that poly-poly(b) coefficients are separated. Identify the magnitude of the isolated coefficients from Round 1, outputing the b largest.
  - Key to the proof: taking the b largest approximate coefficients is a good approximation to the true b largest.
- Execute the two rounds in parallel.

Combinatorial tools

- K-separating sets $S = \{S_1, \ldots S_l\}$. $l=O(k \log^2 n)$

$$X \subset [n], |X| \le k, \exists S_i \in S, \ |S_i \cap X| = 1$$

- K-strongly separating sets $S=\{S_1 \ldots S_m\}$
  $m=O(k^2 \log^2 n)$

$$X \subset [n], |X| \le k, \forall x \in X, \exists S_i \in S, \ S_i \cap X = \{x\}$$

- Hamming matrix H, is $(1+\log n)$ X n
  (H represents 2-separating sets)

1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0

Key elements of the proof

- Over whole class, worst case error is $C_p b^{1-2/p} = \|C_b^{opt}\|_2^2$
- The tail sum after removing the top k' obeys
    - $\sum_{i=b'+1}^{n} |\theta_i| \leq O(b^{1-1/p})$
- Picking $b' > (b\varepsilon^{-p})^{1/(1-p)^2}$ ensures that even if every coefficient after the b' largest is placed in the same set as θ_i, for i in top b, we will recover i.
- Build a b' strongly separating set S, and measure $\chi_S \otimes H$ to identify a superset of the top-b.
- Build a $b'' = (b' \log n)^2$ strongly separating set R, and measure χ_R to allow estimates to be made
- we estimate θ_i using θ'_I: $(\theta'_i - \theta_i)^2 <= \varepsilon^2/(25b) \|C_b^{opt}\|_2^2$

Compressed Sensing Postcursors

- For exponentially decaying compressible families, better constructions $O(b^2 \log)$
- For zero-error case (b-sparse case), Indyk has $O(b \text{ polylog})$.
- For existential **T**, improved decoding time by Gilbert, Strauss, Tropp and Vershynin [06].
- Relationship between Compressed Sensing and Johnson-Lindenstauss, by Baranuik, DeVore et al. [06].

Modern Version 2: Nonuniform Sparse Approximation

- Given dictionary D of N dimensional vectors that span R^N. Also, given an importance (workload) function $\pi[1..N]$.
- Query is a vector **A**[1,…,N].
- Sparse representation for **A** using B terms of D:

$$\overline{\mathrm{A}} = \sum_{d_i \in D,\, i \in \Lambda,\, |\Lambda| \leq b} \alpha_i \, d_i$$

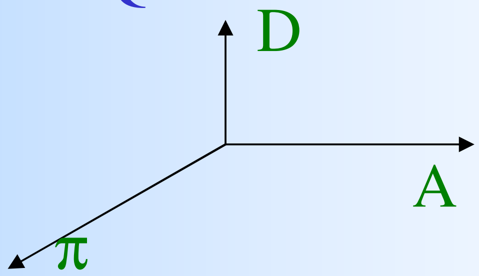
- Minimize

$$\varepsilon_\pi = \sum_j \pi[\mathrm{j}] \; (\mathrm{A[j]} - \overline{\mathrm{A[j]}})^2$$

MAX and other non-L_2 norms: Harb and Guha06

Motivation: Database operations

- Databases work very hard to keep track of the workload (SQL Logs, Index access logs etc.) and use the statistics to optimize the database structure, physical implementation, caching, query optimization, etc.
- Two examples of commercial database engines that use workload information:
  - LEO in DB2 is the recent upgrade from IBM.
  - Self-tuning in Microsoft SQL Server.
- Nontrivial Problem:



Nonuniform SA: Some Results

- Haar dictionary. Sparse representation for **A** using B terms:

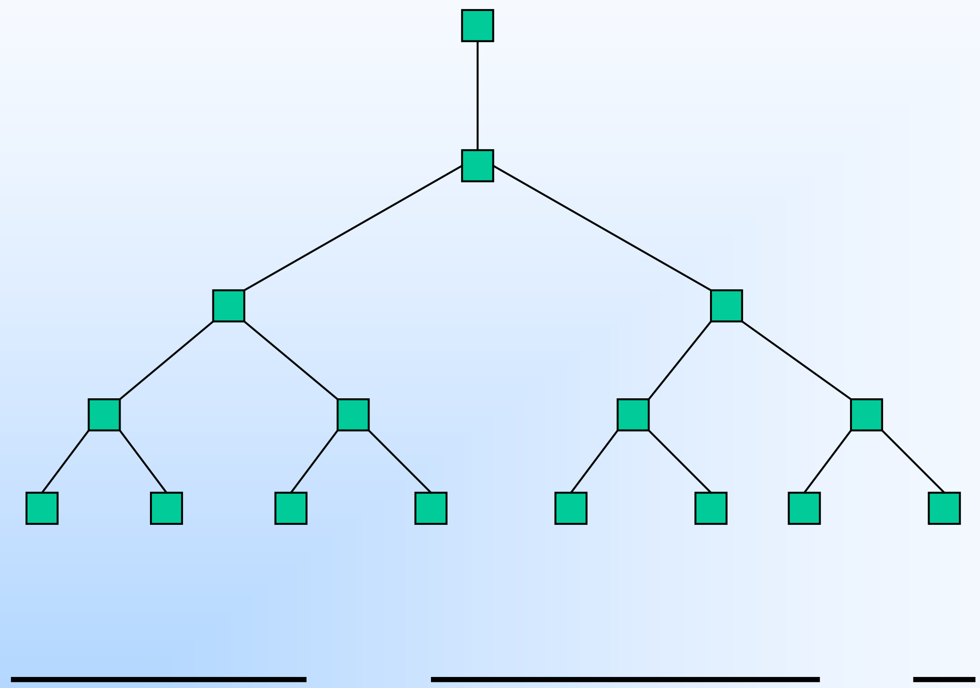
$$\overline{\mathrm{A}} = \sum_{\psi_i \in D,\, i \in \Lambda,\, |\Lambda| \le B} \langle \mathrm{A}, \psi_i \rangle \, \psi_i$$

Minimize error

$$\varepsilon_\pi = \sum_j \pi[j] \; (A[j] - (\sum_{i \in \Lambda} \langle \mathbf{A}, \psi_i \rangle)[j])^2$$

- [Garofalakis+Kumar,Mu,Guha] There exists an $O(N^2 b)$ time algorithm that finds the optimal b Haar wavelets for any function A with importance π.
- If π is a k-piecewise constant function, then the running time is roughly O(n k b).
  - Local Parseval's.

Piecewise Constant π's



Why piecewise constant π's?

- Compressible.
- Good fit for Haar wavelet dictionary.

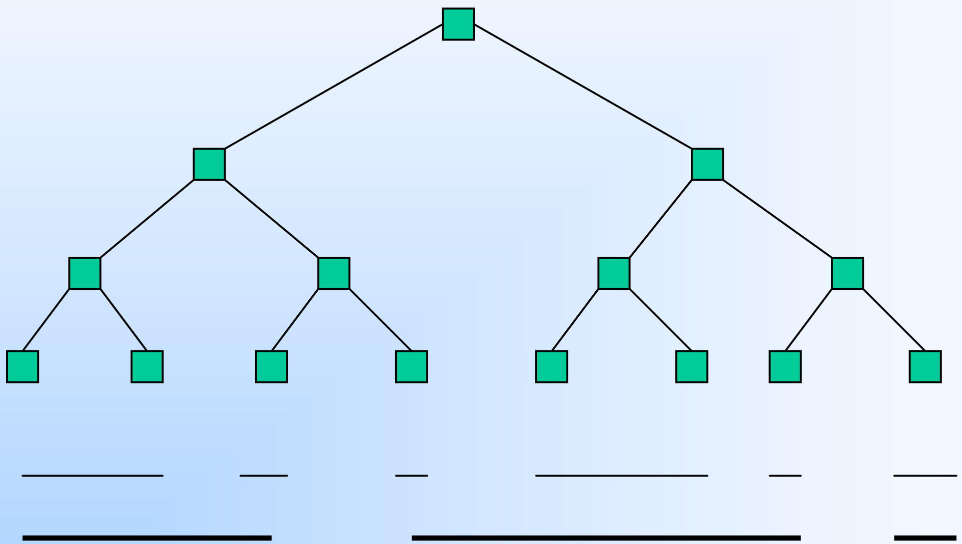
Piecewise constant π

- Say π is disjoint union of k piecewise-constant functions.
- Rewrite it as $O(k \log N)$ dyadic piecewise-constant functions.
- Algorithm: Compressed trie on these intervals.
  - For all internal nodes, do external search.
  - Each leaf corresponds to a dyadic piecewise constant interval: Local Parseval's applies. So pick the best b (except the average).
- [M05] Combining, running time is $O(n\, k\, b)$ since the dyadic trie has $O(k \log N)$ internal nodes. If k dyadic piecewise constant π, then running time is

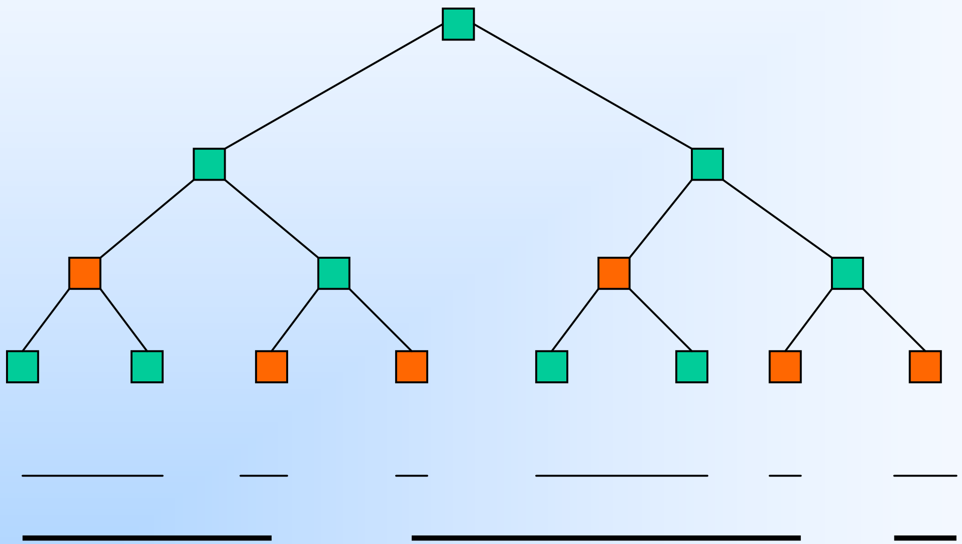
$$O(N + k\, b\, 2^{\min\{k, \log N\}})$$

Near-linear time.

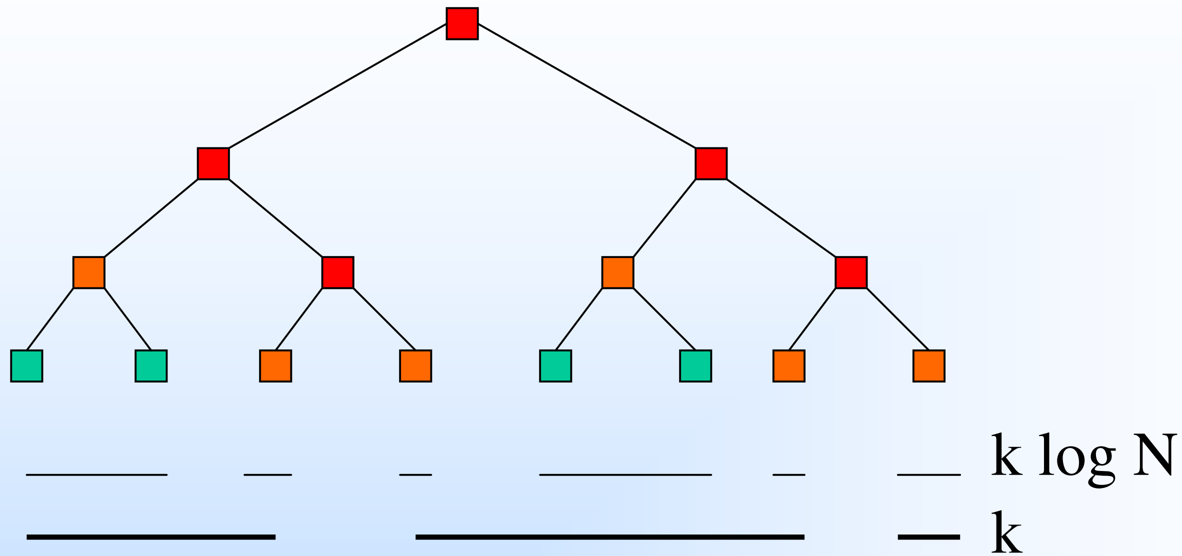
Piecewise Constant π's



Piecewise Constant π's



Piecewise Constant π's



Dynamic Programming at ■

Local Parseval's at ■

Key is Local Parseval's: At any node orange, one can pick the q largest coefficients from wavelet vectors with support contained in its subtree, for any q. Note that this collection of vectors is deficient.

Modern Version: Incoherent Dictionary

- Incoherent dictionary D satisfies $|\langle u.v\rangle| \leq \mu$, for all vectors u, v in D.
- [Gilbert,M,Strauss03, Gilbert,M,Strauss,Tropp04] If $\mu = O(\varepsilon/b^2)$, we can get $1+\varepsilon$ approximation to the best signal representation in b terms in near linear time, after polynomial preprocessing.

Conclusions & Open Problems

- Algorithmic theory of sparse approximation problems. NSF FRG.

- Open problems:
  - Compressed sensing and beyond. Universal decoding.
  - Nonuniform optimization with fourier and wavelets.
  - Incoherent dictionaries: combinations of two basis.
  - Compressed sensing version of matrix approximation.
  - Applications and experiments.

Massive Data Analysis Lab

- MassDAL manages massive streams during the entire lifecycle of data: collect, clean, analyze and integrate into applications.
  http://www.cs.rutgers.edu/~muthu/massdal.html
- I consult for highly specialized, domain-specific data analysis: Patent data analysis, Cellphone call detail records analysis, Auto-insurance fraud, Epidemiology and data mining.

NARUS
Networks

Additional Slides for Nonuniform Sparse Approximation

Approaches to Nonuniform SA

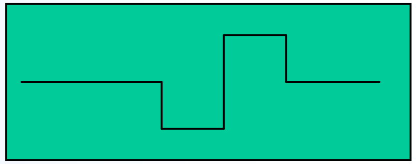
- Parseval's does not help. $\varepsilon_\pi = \sum_j \pi[j](A[j] - (\sum_{i \in \Lambda} \langle \mathbf{A}, \psi_i \rangle)[j])^2$

$$\sum_i \mathbf{A}[i]^2 = \sum_j \langle \mathbf{A}, \psi_j \rangle^2 \qquad \sum_i \pi[i]\mathbf{A}[i]^2 \text{ ??}$$

- Change signal and rewrite things to get Parseval's.

$$\tilde{\mathrm{A}} = \sqrt{\Pi}\,\mathrm{A}$$

$$(\mathrm{A} - \overline{\mathrm{A}})^T \Pi (\mathrm{A} - \overline{\mathrm{A}}) = (\tilde{\mathrm{A}} - \overline{\tilde{\mathrm{A}}})^T (\tilde{\mathrm{A}} - \overline{\tilde{\mathrm{A}}})$$



- Consider weighted versions of Haar [Matias et al. 04]:
  - For ψ_i let l_i (r_i) be sum of $\pi[j]$'s under positive (negative) parts. Mult positive and negative parts by

$$x_i = \sqrt{\frac{r_i}{l_i r_i + l_i^2}} \qquad y_i = \sqrt{\frac{l_i}{l_i r_i + r_i^2}}$$

Both O(N) time.

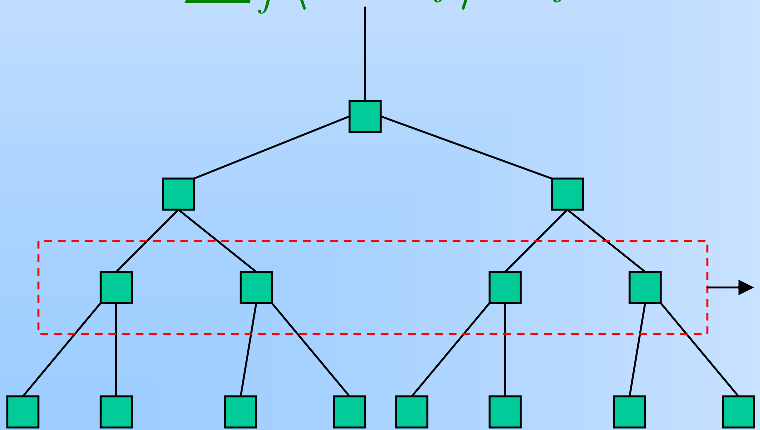
Wavelets: Haar Wavelets (1910)

A[1] A[2] A[3] A[4] A[5] A[6] A[7] A[8]

$\frac{A[1]-A[2]}{\sqrt{2}}$ $\frac{A[3]-A[4]}{\sqrt{2}}$ $\frac{A[5]-A[6]}{\sqrt{2}}$ $\frac{A[7]-A[8]}{\sqrt{2}}$ $\frac{A[1]+A[2]}{\sqrt{2}}$ $\frac{A[3]+A[4]}{\sqrt{2}}$ $\frac{A[5]+A[6]}{\sqrt{2}}$ $\frac{A[7]+A[8]}{\sqrt{2}}$

- N wavelet vectors.
- Orthonormal Basis.

$$\mathbf{A} = \sum_j \langle \mathbf{A}, \psi_j \rangle \, \psi_j$$

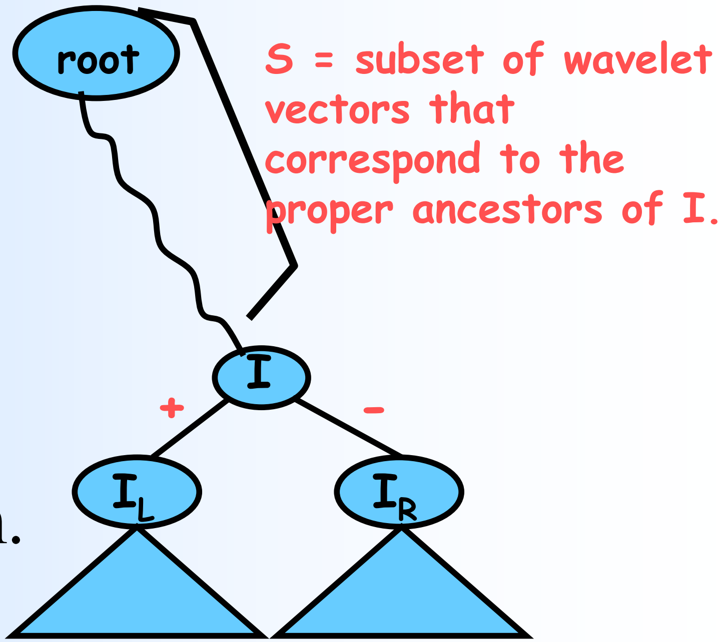


Approach

- Dyadic interval I (non-overlapping partition of 1…N into 2^i sized intervals, for each i).

- E(I,k,S): Minimum error for representing A[I] with importance π[I] using
  - Set S of wavelet vectors whose support contains I,
  - At most k wavelet vectors whose support is contained in I.

- E([1..N],B,ϕ) solves our problem.



Dynamic Programming

- Details:

$$E(I,k,S) = \min_{|\Lambda|\le k;\, \mathrm{supp}(\psi)\subset I} \sum_{j\in I} \pi[j]\,(\mathrm{A}[j] - v - (\sum_{\psi\in\Lambda} \langle \mathrm{A},\psi\rangle \psi)[j])^2$$

$$v[j] = (\sum_{\psi\in S} \langle \mathrm{A},\psi\rangle \psi)[j]$$

- Dynamic Programming:

$$E(I,k,S) = \min_{0\le k'\le k-1} E(I_L, k', S\cup\psi_I) + E(I_R, k-1-k', S\cup\psi_I)$$

$$E(I,k,S) = \min_{0\le k'\le k} E(I_L, k', S) + E(I_R, k-k', S)$$

Complexity

- Dynamic Programming $E(I,k,S)$:
  - Number of problems is $O(N\ b\ N)$ since
    - At most N dyadic intervals.
    - At most $\log N$ wavelet vectors contain a dyadic I.
  - Each problem takes time $O(b)$ to solve.
  - Total time is $O(N^2\ b^2)$.
- Two challenges:
  - Running time is too large.
  - Why use Haar for arbitrary π's?