Algorithms for sparse analysis
Lecture I: Background on sparse approximation

Anna C. Gilbert

Department of Mathematics
University of Michigan

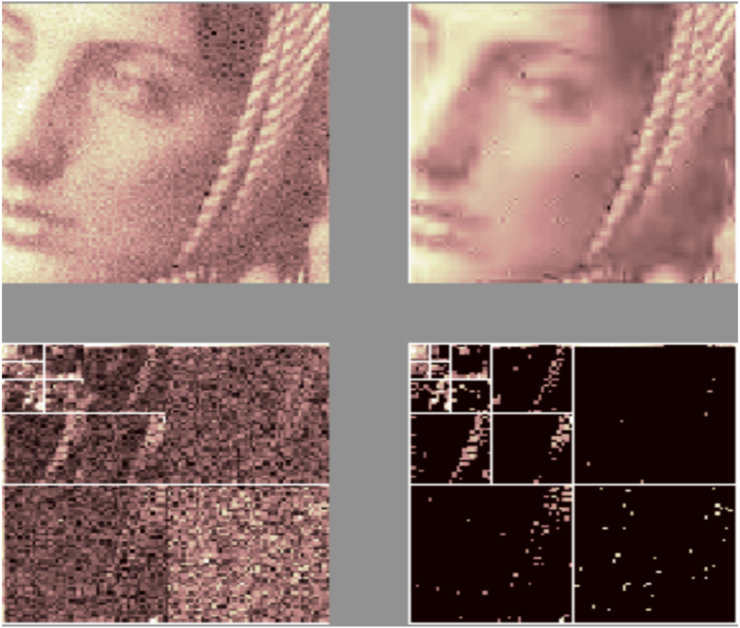
Tutorial on sparse approximations and algorithms

- **Compress** data
  - accurately
  - concisely
  - efficiently (encoding and decoding)
- Focus on
  - mathematical and algorithmic theory: sparse approximation and compressive sensing
  - algorithms: polynomial time, randomized, sublinear, approximation
  - bridge amongst engineering applications, mathematics, and algorithms
- *Not* focus on
  - applications
  - image reconstruction or inverse problems
  - image models, codecs
  - dictionary design

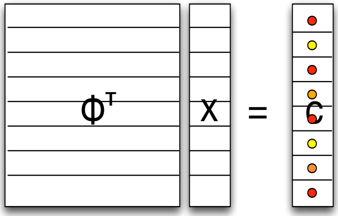
Lectures

- Lecture 1: Background, problem formulation
- Lecture 2: Computational complexity
- Lecture 3: Geometry of dictionaries, greedy algorithms, convex relaxation
- Lecture 4: Sublinear (approximation) algorithms

Basic image compression: transform coding

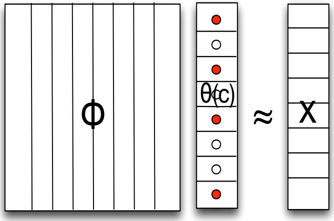


Orthogonal basis Φ: Transform coding

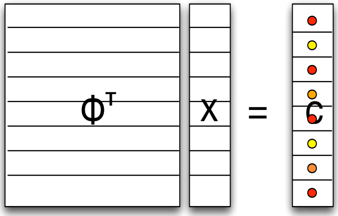


- Compute orthogonal transform

$$\Phi^T x = c$$

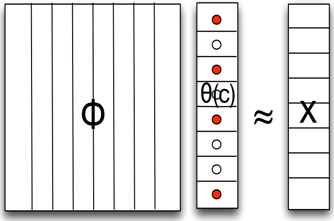


Orthogonal basis Φ: Transform coding



- Compute orthogonal transform

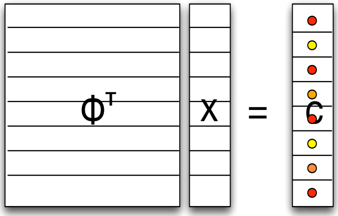
$$\Phi^T x = c$$

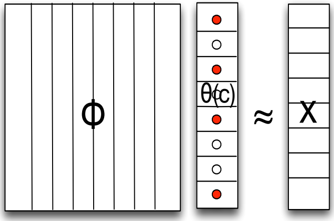


- Threshold small coefficients

$$\Theta(c)$$

Orthogonal basis Φ: Transform coding





- Compute orthogonal transform

$$\Phi^T x = c$$

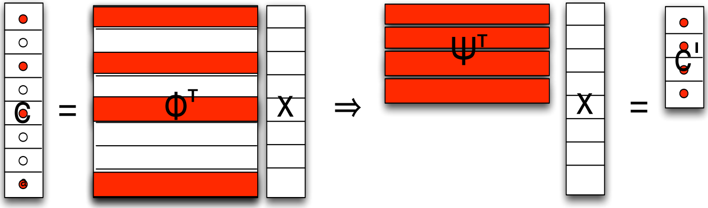
- Threshold small coefficients

$$\Theta(c)$$

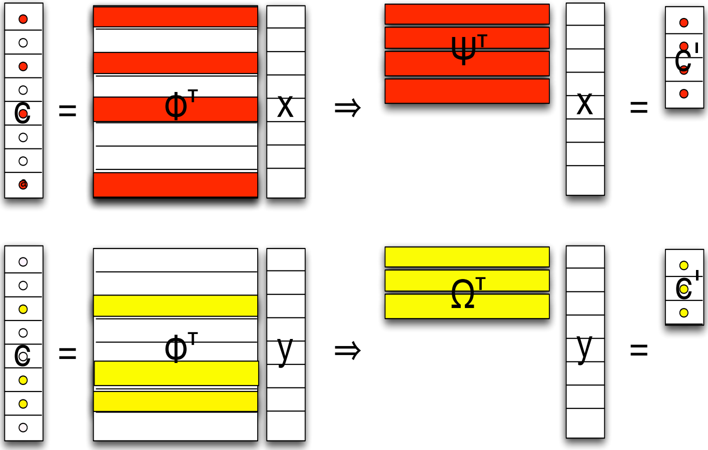
- Reconstruct approximate image

$$\Phi(\Theta(c)) \approx x$$

Nonlinear encoding

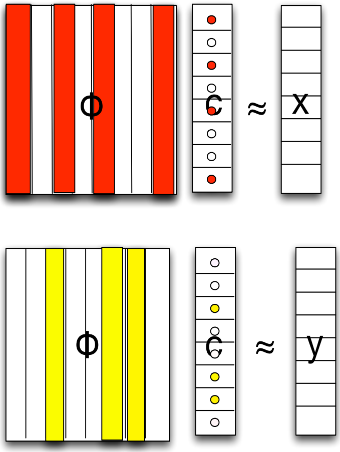


Nonlinear encoding



- Position of nonzeros depends on signal
- Different matrices Φ^T, Ω^T for 2 different signals
- *Adaptive procedure*, adapt for each input

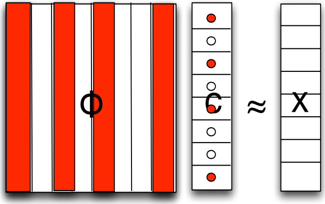
Linear decoding

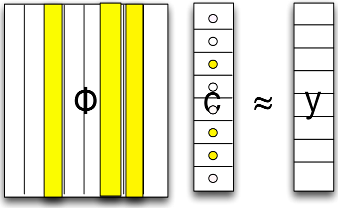


- Given the vector of coefficients and the nonzero positions, recover (approximate) signal via linear combination of coefficients and basis vectors

$$\Phi(\Theta(c)) \approx x$$

Linear decoding



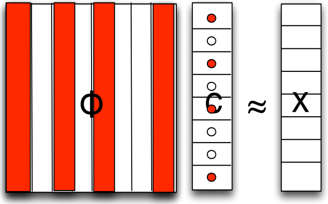


- Given the vector of coefficients and the nonzero positions, recover (approximate) signal via linear combination of coefficients and basis vectors

$$\Phi(\Theta(c)) \approx x$$

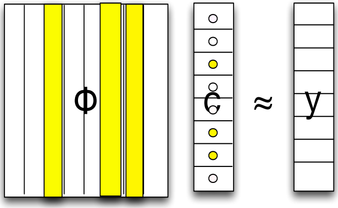
- Decoding procedure *not* signal dependent

Linear decoding



- Given the vector of coefficients and the nonzero positions, recover (approximate) signal via linear combination of coefficients and basis vectors

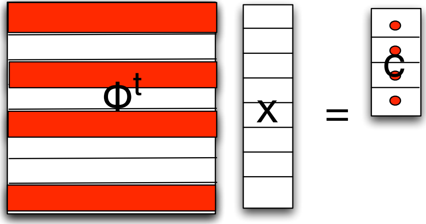
$$\Phi(\Theta(c)) \approx x$$

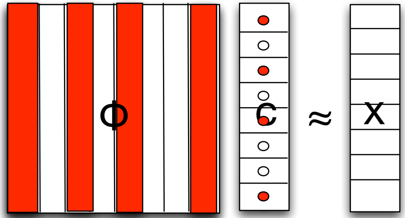


- Decoding procedure *not* signal dependent
- Matrix Φ same for all signals

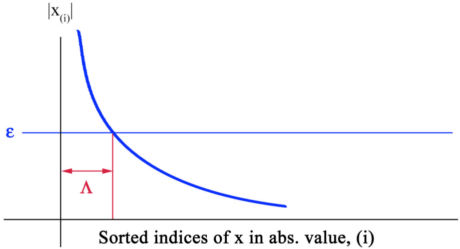
Nonlinear encoding/linear decoding

- Nonlinear encoding
    - **Instance optimal:** given signal x, find best set of coefficients for given signal
    - Algorithm does not guarantee *basis* is a good one, only *representation* good wrt basis
    - Relatively easy to compute
    - *But* compressibility, sparsity of signal depends on choice of orthonormal basis
    - Hard to design *best* orthonormal basis
- Linear decoding: easy to compute





Sparsity/Compressibility



- **Sparsity:** ℓ_0 "norm" = number of non-zero coefficients
- **Compressibility:** rate of decay of sorted coefficients
- Reflects *sparsity/compressibility* of signal in particular basis
- Fewer non-zero coefficients $\Longrightarrow$ fewer items to encode $\Longrightarrow$ fewer total bits to represent signal
- Occam's razor: capture essential features of signal

Basic compressibility results

- Definition
  The optimal k-term error for $x \in \mathbb{R}^N$

$$\sigma_k(x)_p = \min_{z \text{ } k\text{-sparse}} \|x - z\|_p.$$

  The ℓ_p-error of best k-term representation for x in canonical basis.
  - How to relate $\|x\|_p$ to $\sigma_k(x)_p$? restrict $1 \leq p < \infty$.
- Proposition
  If $1 \leq p < \infty$ and $q = (r + \frac{1}{p})^{-1}$, then for all $x \in \mathbb{R}^N$,

$$\sigma_k(x)_p \leq k^{-r} \|x\|_q.$$

Basic compressibility results

- Definition

  The optimal k-term error over the class $K \subset \mathbb{R}^N$

  $$\sigma_k(K)_p = \sup_{x \in K} \sigma_k(x)_p.$$

  - Let $K = B_q^N = \{x \in \mathbb{R}^N \mid \|x\|_q = 1\}$.

    $$\sigma_k(K)_p \le k^{-r}$$

    with $r = \frac{1}{q} - \frac{1}{p}$.

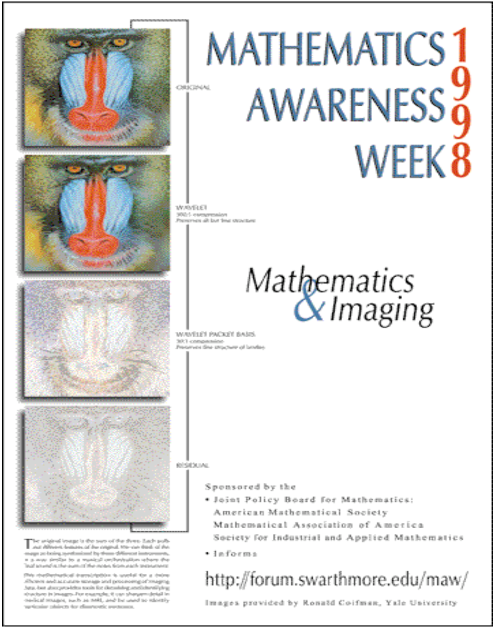
Redundancy

If one orthonormal basis is good, surely two (or more) are better...

Redundancy

If one orthonormal basis is good, surely two (or more) are better...

...especially for images



MATHEMATICS AWARENESS WEEK 1998
ORIGINAL
WAVELET
WAVELET PACKET BASIS
RESIDUAL
Mathematics & Imaging
Sponsored by the
• Joint Policy Board for Mathematics:
American Mathematical Society
Mathematical Association of America
Society for Industrial and Applied Mathematics
• Informs
http://forum.swarthmore.edu/maw/
Images provided by Ronald Coifman, Yale University

Dictionary

Definition

A dictionary Φ *in* $\mathbb{R}^d$ *is a collection* $\{\varphi_\ell\}_{\ell=1}^N \subset \mathbb{R}^d$ *of unit-norm vectors:* $\|\varphi_\ell\|_2 = 1$ *for all* ℓ*.*

- Elements are called *atoms*
- If $\operatorname{span}\{\varphi_\ell\} = \mathbb{R}^d$, the dictionary is *complete*
- If $\{\varphi_\ell\}$ are linearly dependent, the dictionary is *redundant*

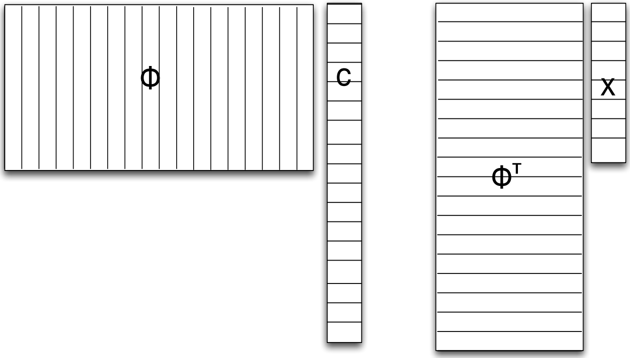
Matrix representation

Form a matrix

$$\Phi = \begin{bmatrix} \varphi_1 & \varphi_2 & \dots & \varphi_N \end{bmatrix}$$

so that

$$\Phi c = \sum_{\ell} c_\ell \varphi_\ell .$$

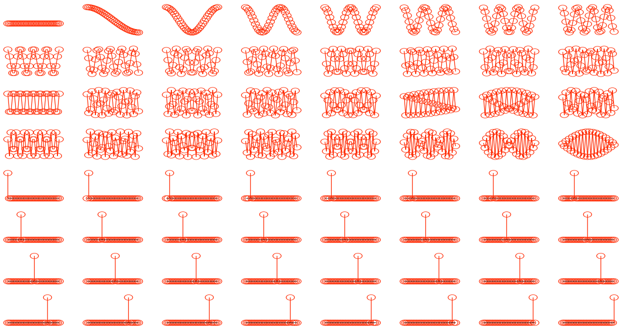


Examples: Fourier—Dirac

$\Phi = [\,\mathcal{F}\,|\,I\,]$

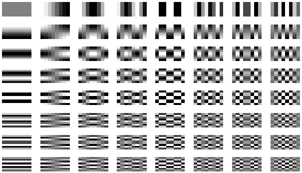
$$\varphi_\ell(t) = \frac{1}{\sqrt{d}} \mathrm{e}^{2\pi \mathrm{i} \ell t/d} \quad \ell = 1, 2, \ldots, d$$

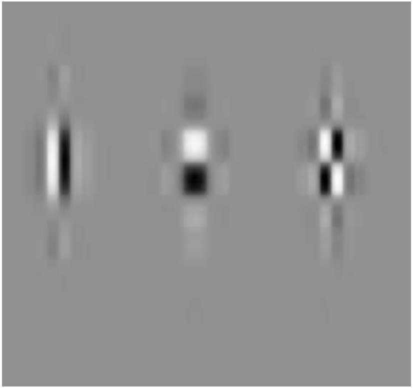
$$\varphi_\ell(t) = \delta_\ell(t) \quad \ell = d+1, d+2, \ldots, 2d$$



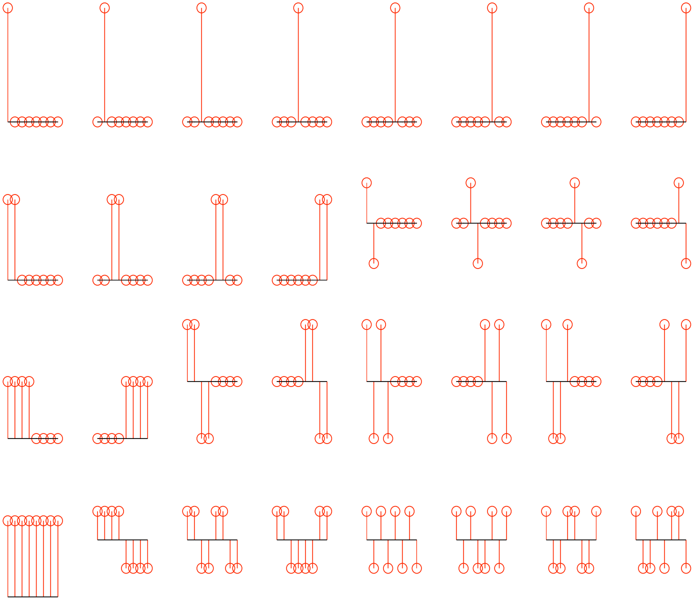
Examples: DCT—Wavelets, 2 dimensions

$$\Phi = [\,\mathcal{F}\,|\,\mathcal{W}\,]$$

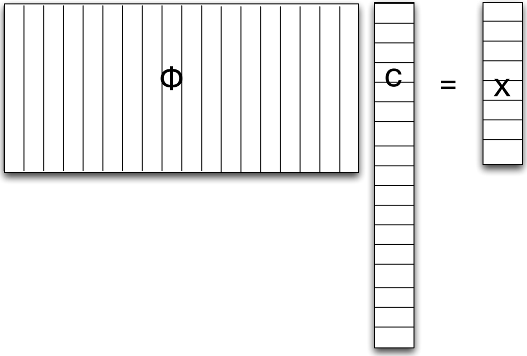




Examples: Wavelet packets



Sparsity: Tradeoff



cost of representation $\longleftrightarrow$ error in approximation

rate $\longleftrightarrow$ **distortion**

Sparse Problems

- Exact. Given a vector $x \in \mathbb{R}^d$ and a complete dictionary Φ, solve

$$\arg\min_c \|c\|_0 \quad \text{s.t.} \quad x = \Phi c$$

i.e., find a sparsest representation of x over Φ.

Sparse Problems

- Exact. Given a vector $x \in \mathbb{R}^d$ and a complete dictionary Φ, solve

$$\arg\min_c \|c\|_0 \quad \text{s.t.} \quad x = \Phi c$$

  i.e., find a sparsest representation of x over Φ.

- Error. Given $\epsilon \geq 0$, solve

$$\arg\min_c \|c\|_0 \quad \text{s.t.} \quad \|x - \Phi c\|_2 \leq \epsilon$$

  i.e., find a sparsest approximation of x that achieves error ϵ.

SPARSE Problems

- EXACT. Given a vector $x \in \mathbb{R}^d$ and a complete dictionary Φ, solve

$$\arg\min_c \|c\|_0 \quad \text{s.t.} \quad x = \Phi c$$

i.e., find a sparsest representation of x over Φ.

- ERROR. Given $\epsilon \geq 0$, solve

$$\arg\min_c \|c\|_0 \quad \text{s.t.} \quad \|x - \Phi c\|_2 \leq \epsilon$$

i.e., find a sparsest approximation of x that achieves error ϵ.

- SPARSE. Given $k \geq 1$, solve

$$\arg\min_c \|x - \Phi c\|_2 \quad \text{s.t.} \quad \|c\|_0 \leq k$$

i.e., find the best approximation of x using k atoms.

Computational complexity

- How hard is it to solve these problems?
- How difficult are these problems as compared to well-studied problems; e.g., PRIMES, SAT?
- What is the asymptotic complexity of the problem?
- Is there a feasible algorithm that is guaranteed to solve the problem?

- Sparse approximation problems are at least as hard as SAT

Complexity theory: Decision problems

- Fundamental problem type: output is YES or NO

- RELPRIME. Are a and b relatively prime?
- SAT. Boolean expression . Given a Boolean expression with AND, OR, NOT, variables, and parentheses only, is there some assignment of TRUE and FALSE to variables that makes expression TRUE?

- D-EXACT. Given a vector $x \in \mathbb{R}^d$, a complete $d \times N$ dictionary Φ, and a sparsity parameter k, does there exist a vector $c \in \mathbb{R}^N$ with $\|c\|_0 \leq k$ such that $\Phi c = x$?

- Can be used as a subroutine (partially) to solve EXACT

Complexity theory: Languages

- Encode input as a finite string of 0s and 1s

- Definition
  A language L is a subset of the set of all (finite length) strings.

- Decision problem = language L
- Given input x, decide if $x \in L$ or if $x \notin L$

$RelPrime = \{$binary encodings of pairs (a,b) s.t. $\gcd(a,b) = 1\}$

- Algorithm A_L for deciding L

$$A_L(x) = \begin{cases} Yes & \text{iff } x \in L \\ No & \text{otherwise} \end{cases}$$

Complexity theory: **P**

- **P** = polynomial time

- Definition
  P *is the class of languages L that are decidable in polynomial time; i.e., there is an algorithm A_L such that*
    - $x \in L$ *iff* $A_L(x) = Yes$ *(otherwise* $A_L(x) = No$*)*
    - *there is some n so that for all inputs x, the running time of* A_L *on x is less than* $|x|^n$

Complexity theory: **NP**

NP $\neq$ "Not Polynomial time"

Complexity theory: Verify vs. determine

- *Verifying* existence of k-sparse vector c with $\Phi c = x$ easier than *determining* existence
- Given witness c, check (i) $\|c\|_0 = k$ and (ii) $\Phi c = x$
- Checks can be done in time polynomial in size of Φ, x, k

- Definition

  A language L has a (polynomial time) verifier if there is an algorithm V such that

$$L = \{x \,|\, \exists\, w \text{ s.t. } V \text{ accepts } (x, w)\}$$

(and algorithm V runs in time polynomial in the length of x).

NP

- **NP** = nondeterministic polynomial time

Definition

NP *is the class of languages* L *that have polynomial time verifiers.*

NP

- **NP** = nondeterministic polynomial time

Definition

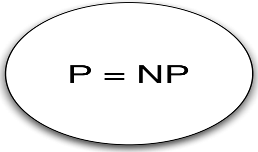
NP *is the class of languages L that have polynomial time verifiers.*

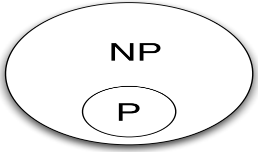
- **P** = membership *decided* efficiently
- **NP** = membership *verified* efficiently
- **P** = **NP**?

NP

- **NP** = nondeterministic polynomial time

Definition
NP *is the class of languages L that have polynomial time verifiers.*





- **P** = membership *decided* efficiently
- **NP** = membership *verified* efficiently
- **P** = **NP**?

Summary

- Formal, precise definitions of sparse approximation problems
- Set up complexity theory: goals, definitions, 2 complexity classes
- **Next lecture:** reductions amongst problem, hardness results