Recent Theoretical Advances in Sparse Approximation

Joel A. Tropp

<jtropp@ices.utexas.edu>

Institute for Computational Engineering and Sciences

The University of Texas at Austin

Includes joint work with A. C. Gilbert, S. Muthukrishnan and M. J. Strauss of AT&T Research. S. Muthukrishnan is also affiliated with Rutgers Univ.

What is Sparse Approximation?

- We work in the finite-dimensional Hilbert space $\mathbb{C}^d$
- Let $\mathscr{D} = \{\boldsymbol{\varphi}_\omega\}$ be a *dictionary* of N unit-norm *atoms* indexed by Ω
- Let m be a fixed, positive integer
- Suppose $\boldsymbol{x}$ is an *arbitrary* input vector

What is Sparse Approximation?

- We work in the finite-dimensional Hilbert space $\mathbb{C}^d$
- Let $\mathscr{D} = \{\boldsymbol{\varphi}_\omega\}$ be a *dictionary* of N unit-norm *atoms* indexed by Ω
- Let m be a fixed, positive integer
- Suppose $\boldsymbol{x}$ is an *arbitrary* input vector
- The *sparse approximation problem* is to solve

$$\min_{\Lambda \subset \Omega} \quad \min_{\boldsymbol{b} \in \mathbb{C}^\Lambda} \quad \left\| \boldsymbol{x} - \sum_{\lambda \in \Lambda} b_\lambda \, \boldsymbol{\varphi}_\lambda \right\|_2 \qquad \text{subject to} \qquad |\Lambda| \leq m$$

- The inner minimization is a least squares problem
- But the outer minimization is *combinatorial*

What is Sparse Approximation?

- We work in the finite-dimensional Hilbert space $\mathbb{C}^d$
- Let $\mathscr{D} = \{\boldsymbol{\varphi}_\omega\}$ be a *dictionary* of N unit-norm *atoms* indexed by Ω
- Let m be a fixed, positive integer
- Suppose $\boldsymbol{x}$ is an *arbitrary* input vector
- The *sparse approximation problem* is to solve

$$\min_{\Lambda \subset \Omega} \quad \min_{\boldsymbol{b} \in \mathbb{C}^\Lambda} \quad \left\| \boldsymbol{x} - \sum_{\lambda \in \Lambda} b_\lambda \, \boldsymbol{\varphi}_\lambda \right\|_2 \qquad \text{subject to} \qquad |\Lambda| \leq m$$

- The inner minimization is a least squares problem
- But the outer minimization is *combinatorial*
- Formally, we call the problem $(\mathscr{D}, m)$-$\textsc{Sparse}$

Basic Dictionary Properties

- The dictionary is *complete* if the atoms span $\mathbb{C}^d$
- The dictionary is *redundant* if it contains linearly dependent atoms

Basic Dictionary Properties

- The dictionary is *complete* if the atoms span $\mathbb{C}^d$
- The dictionary is *redundant* if it contains linearly dependent atoms

- A complete dictionary can represent every vector without error
- Each vector has infinitely many representations over a redundant dictionary

Basic Dictionary Properties

- The dictionary is *complete* if the atoms span $\mathbb{C}^d$
- The dictionary is *redundant* if it contains linearly dependent atoms

- A complete dictionary can represent every vector without error
- Each vector has infinitely many representations over a redundant dictionary

- In most modern applications, dictionaries are complete and redundant

Subset Selection in Regression

- Suppose $\boldsymbol{x}$ is a vector of d observations of a random variable $\boldsymbol{X}$
- Suppose $\boldsymbol{\varphi}_\omega$ is a vector of d observations of random variable $\boldsymbol{\Phi}_\omega$
- Want to find a small subset of $\{\boldsymbol{\Phi}_\omega\}$ for linear prediction of $\boldsymbol{X}$

Subset Selection in Regression

- Suppose $\boldsymbol{x}$ is a vector of d observations of a random variable $\boldsymbol{X}$
- Suppose $\boldsymbol{\varphi}_\omega$ is a vector of d observations of random variable $\boldsymbol{\Phi}_\omega$
- Want to find a small subset of $\{\boldsymbol{\Phi}_\omega\}$ for linear prediction of $\boldsymbol{X}$
- *Method:* Solve the sparse approximation problem!

Subset Selection in Regression

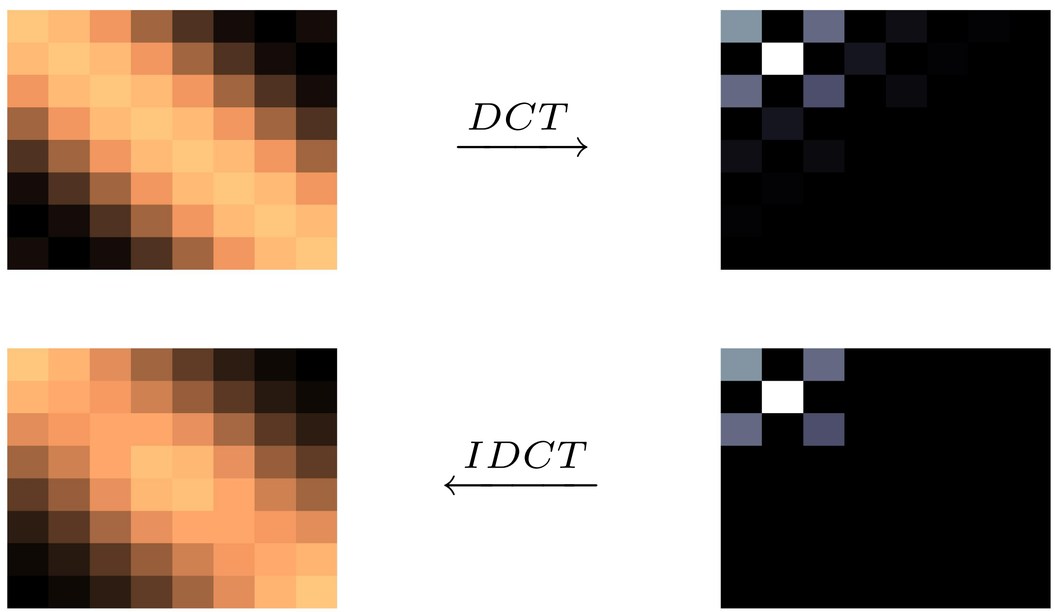
- Suppose $\boldsymbol{x}$ is a vector of d observations of a random variable $\boldsymbol{X}$
- Suppose $\boldsymbol{\varphi}_\omega$ is a vector of d observations of random variable $\boldsymbol{\Phi}_\omega$
- Want to find a small subset of $\{\boldsymbol{\Phi}_\omega\}$ for linear prediction of $\boldsymbol{X}$
- *Method:* Solve the sparse approximation problem!

- Statisticians have developed many approaches
  1. Forward selection
  2. Backward elimination
  3. Sequential replacement
  4. Stepwise regression [Efroymson 1960]
  5. Exhaustive search [Garside 1965, Beale et al. 1967]
  6. Projection Pursuit Regression [Friedman–Stuetzle 1981]

Reference: [A. J. Miller 2002]

Transform Coding

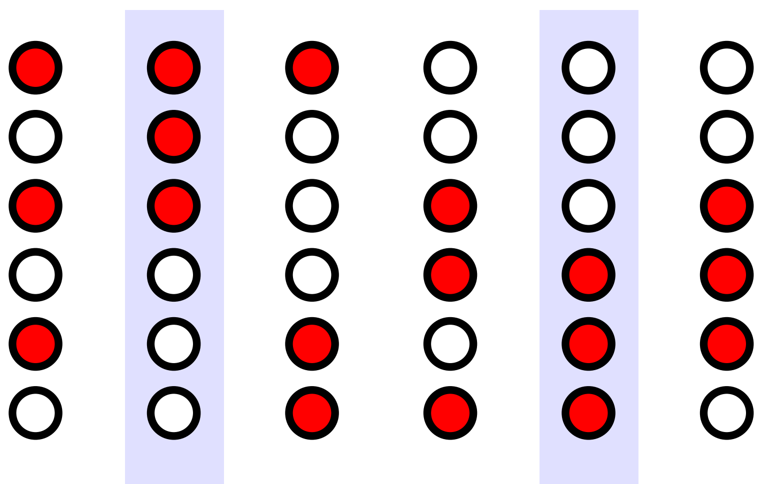
- In simplest form, can be viewed as a sparse approximation problem



Reference: [Evans-Mersereau 2003]

Computational Complexity

Theorem 1. [Davis (1994), Natarajan (1995)] *Any instance of Exact Cover by Three Sets* (X3C) *is reducible in polynomial time to a sparse approximation problem.*



An instance of X3C

Computational Complexity II

Corollary 2. *Any algorithm that can solve* $(\mathscr{D}, m)$-SPARSE *for* every *dictionary and sparsity level must solve an NP-hard problem.*

- It is widely believed that no tractable algorithms exist for NP-hard problems

Computational Complexity II

Corollary 2. *Any algorithm that can solve* $(\mathscr{D}, m)$-SPARSE *for* every *dictionary and sparsity level must solve an NP-hard problem.*

- It is widely believed that no tractable algorithms exist for NP-hard problems

- *BUT* a specific problem $(\mathscr{D}, m)$-SPARSE may be easy
- *AND* preprocessing is allowed

Orthonormal Dictionaries

- Suppose that $\mathscr{D}$ is an orthonormal basis (ONB)

Orthonormal Dictionaries

- Suppose that $\mathscr{D}$ is an orthonormal basis (ONB)
- For any vector $\boldsymbol{x}$ and sparsity level m,
  1. Sort the indices $\{\omega_n\}$ so the numbers $|\langle \boldsymbol{x}, \boldsymbol{\varphi}_{\omega_n} \rangle|$ are decreasing

Orthonormal Dictionaries

- Suppose that $\mathscr{D}$ is an orthonormal basis (ONB)
- For any vector $\boldsymbol{x}$ and sparsity level m,
  1. Sort the indices $\{\omega_n\}$ so the numbers $|\langle \boldsymbol{x}, \boldsymbol{\varphi}_{\omega_n} \rangle|$ are decreasing
  2. The solution to $(\mathscr{D}, m)$-Sparse for input $\boldsymbol{x}$ is

$$\sum_{n=1}^{m} \langle \boldsymbol{x}, \boldsymbol{\varphi}_{\omega_n} \rangle \; \boldsymbol{\varphi}_{\omega_n}$$

Orthonormal Dictionaries

- Suppose that $\mathscr{D}$ is an orthonormal basis (ONB)
- For any vector $\boldsymbol{x}$ and sparsity level m,
  1. Sort the indices $\{\omega_n\}$ so the numbers $|\langle \boldsymbol{x}, \boldsymbol{\varphi}_{\omega_n} \rangle|$ are decreasing
  2. The solution to $(\mathscr{D}, m)$-SPARSE for input $\boldsymbol{x}$ is

$$\sum_{n=1}^{m} \langle \boldsymbol{x}, \boldsymbol{\varphi}_{\omega_n} \rangle \; \boldsymbol{\varphi}_{\omega_n}$$

  3. The squared approximation error is

$$\sum_{n=m+1}^{d} |\langle \boldsymbol{x}, \boldsymbol{\varphi}_{\omega_n} \rangle|^2$$

Orthonormal Dictionaries

- Suppose that $\mathscr{D}$ is an orthonormal basis (ONB)
- For any vector $\boldsymbol{x}$ and sparsity level m,
  1. Sort the indices $\{\omega_n\}$ so the numbers $|\langle \boldsymbol{x}, \boldsymbol{\varphi}_{\omega_n} \rangle|$ are decreasing
  2. The solution to $(\mathscr{D}, m)$-Sparse for input $\boldsymbol{x}$ is

$$\sum_{n=1}^{m} \langle \boldsymbol{x}, \boldsymbol{\varphi}_{\omega_n} \rangle \; \boldsymbol{\varphi}_{\omega_n}$$

  3. The squared approximation error is

$$\sum_{n=m+1}^{d} \left| \langle \boldsymbol{x}, \boldsymbol{\varphi}_{\omega_n} \rangle \right|^2$$

Insight: $(\mathscr{D}, m)$-Sparse can be solved approximately so long as sub-collections of m atoms in $\mathscr{D}$ are sufficiently close to being orthogonal.

Coherence

- Donoho and Huo introduced the *coherence parameter* μ of a dictionary:

$$\mu = \max_{j \neq k} \left| \left\langle \boldsymbol{\varphi}_{\omega_j}, \boldsymbol{\varphi}_{\omega_k} \right\rangle \right|$$

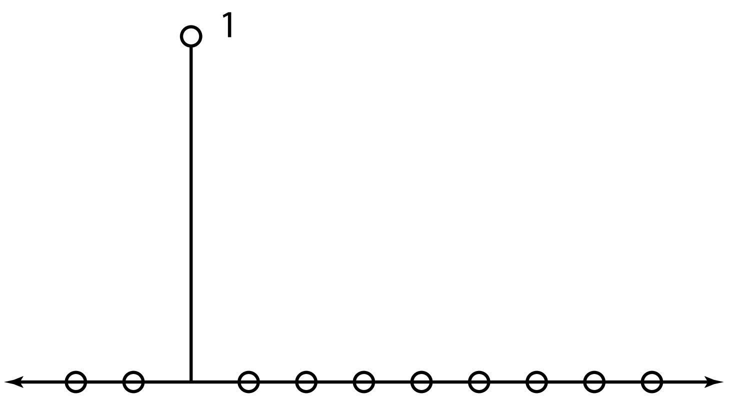
- Measures how much distinct atoms look alike

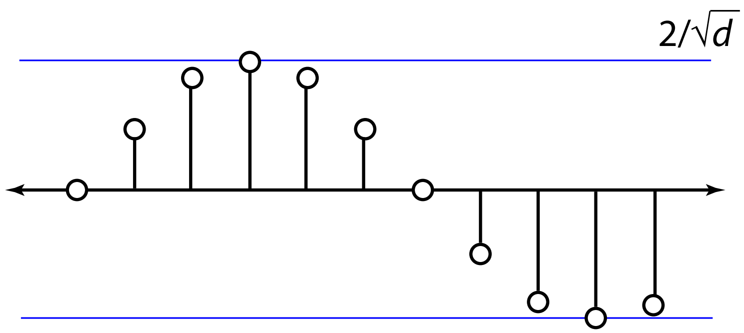
Coherence

- Donoho and Huo introduced the *coherence parameter* μ of a dictionary:

$$\mu = \max_{j \neq k} \left| \left\langle \boldsymbol{\varphi}_{\omega_j}, \boldsymbol{\varphi}_{\omega_k} \right\rangle \right|$$

- Measures how much distinct atoms look alike
- Many natural dictionaries are incoherent [Donoho–Huo 2000]
- Example: Spikes + sines





Coherence Bounds

- In general,

$$\mu \geq \sqrt{\frac{N-d}{d\,(N-1)}}$$

- If the dictionary contains an orthonormal basis,

$$\mu \geq \sqrt{\frac{1}{d}}$$

Coherence Bounds

- In general,

$$\mu \geq \sqrt{\frac{N-d}{d\,(N-1)}}$$

- If the dictionary contains an orthonormal basis,

$$\mu \geq \sqrt{\frac{1}{d}}$$

- Incoherent dictionaries can be *enormous* [GMS 2003]

Quasi-Coherence

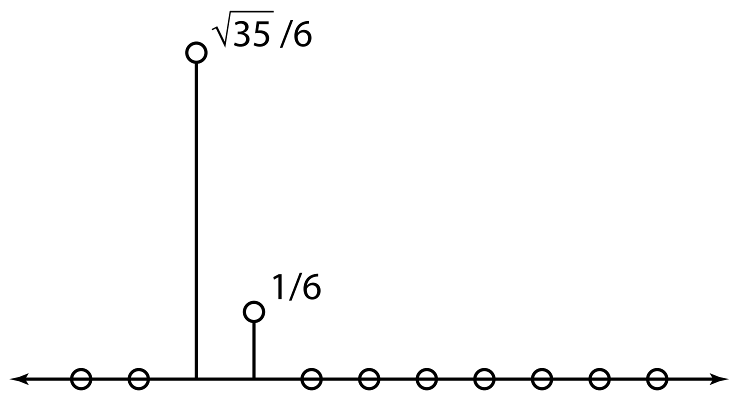
- Donoho–Elad [2003] and JAT [2003] independently introduced the *quasi-coherence*:

$$\mu_1(m) = \max_{\omega} \max_{\lambda_1, \dots, \lambda_m} \sum_{t=1}^{m} |\langle \boldsymbol{\varphi}_{\omega}, \boldsymbol{\varphi}_{\lambda_t} \rangle|$$

- Observe that $\mu_1(1) = \mu$
- Generalizes the cumulative coherence: $\mu_1(m) \le \mu \, m$

Quasi-Coherence Example

- Consider the dictionary of translates of a double pulse:



- The coherence is $\mu = \sqrt{35}/36$
- The quasi-coherence is

$$\mu_1(m) = \begin{cases} \sqrt{35}/36, & m = 1 \\ \sqrt{35}/18, & m = 2 \\ \sqrt{35}/12, & m \geq 3 \end{cases}$$

Roadmap

- First, a few basic algorithms for sparse approximation
- Then, the role of quasi-coherence in the performance of these algorithms
- Finally, a new algorithm that offers better approximation guarantees

Matching Pursuit (MP)

- In 1993, Mallat and Zhang presented a greedy method for sparse approximation over redundant dictionaries
  - Equivalent to Projection Pursuit Regression [Friedman–Stuetzle 1981]
  - Developed independently by Qian and Chen [1993]

Matching Pursuit (MP)

- In 1993, Mallat and Zhang presented a greedy method for sparse approximation over redundant dictionaries
  - Equivalent to Projection Pursuit Regression [Friedman–Stuetzle 1981]
  - Developed independently by Qian and Chen [1993]
- Procedure:
  1. Initialize $\boldsymbol{a}_0 = \mathbf{0}$ and $\boldsymbol{r}_0 = \boldsymbol{x}$

Matching Pursuit (MP)

- In 1993, Mallat and Zhang presented a greedy method for sparse approximation over redundant dictionaries
  - Equivalent to Projection Pursuit Regression [Friedman–Stuetzle 1981]
  - Developed independently by Qian and Chen [1993]
- Procedure:
  1. Initialize $\boldsymbol{a}_0 = \mathbf{0}$ and $\boldsymbol{r}_0 = \boldsymbol{x}$
  2. At step t, select an atom $\boldsymbol{\varphi}_{\lambda_t}$ that solves

$$\max_{\omega} |\langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_{\omega} \rangle|$$

Matching Pursuit (MP)

- In 1993, Mallat and Zhang presented a greedy method for sparse approximation over redundant dictionaries
  - Equivalent to Projection Pursuit Regression [Friedman–Stuetzle 1981]
  - Developed independently by Qian and Chen [1993]
- Procedure:
  1. Initialize $\boldsymbol{a}_0 = \boldsymbol{0}$ and $\boldsymbol{r}_0 = \boldsymbol{x}$
  2. At step t, select an atom $\boldsymbol{\varphi}_{\lambda_t}$ that solves

$$\max_{\omega} |\langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_{\omega} \rangle|$$

  3. Form a new approximation and residual

$$\boldsymbol{a}_t = \boldsymbol{a}_{t-1} + \langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_{\lambda_t} \rangle \, \boldsymbol{\varphi}_{\lambda_t}$$

$$\boldsymbol{r}_t = \boldsymbol{r}_{t-1} - \langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}_{\lambda_t} \rangle \, \boldsymbol{\varphi}_{\lambda_t}$$

Convergence of Matching Pursuit

- Huber [1985] and Jones [1987] developed convergence theory
- Matching Pursuit generates residuals that approach zero:

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq C(\mathscr{D})^m \, \|\boldsymbol{x}\|_2$$

- The constant $C(\mathscr{D})$ is essentially the *covering radius* of the dictionary

Convergence of Matching Pursuit

- Huber [1985] and Jones [1987] developed convergence theory
- Matching Pursuit generates residuals that approach zero:

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq C(\mathscr{D})^m \ \|\boldsymbol{x}\|_2$$

- The constant $C(\mathscr{D})$ is essentially the *covering radius* of the dictionary

- Prove nothing about whether MP solves the sparse problem
- Until recently, this was the only type of result available

Reference: [Temlyakov 2002]

Sparsity Lost

- DeVore and Temlyakov showed that MP may fail to recover a vector with an exact, sparse representation [1996]

Sparsity Lost

- DeVore and Temlyakov showed that MP may fail to recover a vector with an exact, sparse representation [1996]
- Suppose that $\mathscr{D}$ is an orthonormal basis for $\mathbb{C}^d$
- Adjoin the unit-norm vector

$$\boldsymbol{\psi} = \alpha \left[\boldsymbol{\varphi}_1 + \boldsymbol{\varphi}_2 + \sum_{n=3}^{d} \tfrac{1}{(n-2)^2} \, \boldsymbol{\varphi}_n \right]$$

Sparsity Lost

- DeVore and Temlyakov showed that MP may fail to recover a vector with an exact, sparse representation [1996]
- Suppose that $\mathscr{D}$ is an orthonormal basis for $\mathbb{C}^d$
- Adjoin the unit-norm vector

$$\boldsymbol{\psi} = \alpha \left[\boldsymbol{\varphi}_1 + \boldsymbol{\varphi}_2 + \sum_{n=3}^{d} \tfrac{1}{(n-2)^2} \, \boldsymbol{\varphi}_n \right]$$

- Consider the input vector $\boldsymbol{x} = \boldsymbol{\varphi}_1 + \boldsymbol{\varphi}_2$
- MP continues forever with approximation error

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 = O(1/\sqrt{m})$$

Orthogonal Matching Pursuit (OMP)

- Davis, Mallat and Zhang proposed a better greedy method [1997]
  - Originally developed by Chen, Billings and Luo [1989]
  - Also introduced by Pati, Rezaiifar and Krishnaprasad [1993]

Orthogonal Matching Pursuit (OMP)

- Davis, Mallat and Zhang proposed a better greedy method [1997]
  - Originally developed by Chen, Billings and Luo [1989]
  - Also introduced by Pati, Rezaiifar and Krishnaprasad [1993]
- Selects atoms the same way as MP
- Computes new approximation and residual via

$$\boldsymbol{a}_t = \boldsymbol{a}_{t-1} + \left\langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}^{\perp}_{\lambda_t} \right\rangle \boldsymbol{\varphi}^{\perp}_{\lambda_t}$$

$$\boldsymbol{r}_t = \boldsymbol{r}_{t-1} - \left\langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}^{\perp}_{\lambda_t} \right\rangle \boldsymbol{\varphi}^{\perp}_{\lambda_t}$$

Orthogonal Matching Pursuit (OMP)

- Davis, Mallat and Zhang proposed a better greedy method [1997]
  - Originally developed by Chen, Billings and Luo [1989]
  - Also introduced by Pati, Rezaiifar and Krishnaprasad [1993]
- Selects atoms the same way as MP
- Computes new approximation and residual via

$$\boldsymbol{a}_t = \boldsymbol{a}_{t-1} + \left\langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}^{\perp}_{\lambda_t} \right\rangle \boldsymbol{\varphi}^{\perp}_{\lambda_t}$$

$$\boldsymbol{r}_t = \boldsymbol{r}_{t-1} - \left\langle \boldsymbol{r}_{t-1}, \boldsymbol{\varphi}^{\perp}_{\lambda_t} \right\rangle \boldsymbol{\varphi}^{\perp}_{\lambda_t}$$

- Convergence similar to MP but stops after d steps [Temlyakov 2002]
- Counterexamples prove OMP may fail to recover sparse superpositions [Chen–Donoho–Saunders 1999]

ℓ_1 Minimization

- Chen, Donoho and Saunders introduced a more global approach [1999]

ℓ_1 Minimization

- Chen, Donoho and Saunders introduced a more global approach [1999]
- Replace $(\mathscr{D}, m)$-Sparse by a convex relaxation:

$$\min_{\boldsymbol{b} \in \mathbb{C}^N} \quad \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \sum_{\omega \in \Omega} b_\omega \, \boldsymbol{\varphi}_\omega = \boldsymbol{x}$$

- Hope the answers coincide

ℓ_1 Minimization

- Chen, Donoho and Saunders introduced a more global approach [1999]
- Replace $(\mathscr{D}, m)$-SPARSE by a convex relaxation:

$$\min_{\boldsymbol{b} \in \mathbb{C}^N} \quad \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \sum_{\omega \in \Omega} b_\omega \, \boldsymbol{\varphi}_\omega = \boldsymbol{x}$$

- Hope the answers coincide

- Copious numerical evidence that it succeeds for sparse approximation
- Penalized version for de-noising

ℓ_1 Minimization

- Chen, Donoho and Saunders introduced a more global approach [1999]
- Replace $(\mathscr{D}, m)$-SPARSE by a convex relaxation:

$$\min_{\boldsymbol{b} \in \mathbb{C}^N} \quad \|\boldsymbol{b}\|_1 \qquad \text{subject to} \qquad \sum_{\omega \in \Omega} b_\omega \, \boldsymbol{\varphi}_\omega = \boldsymbol{x}$$

- Hope the answers coincide

- Copious numerical evidence that it succeeds for sparse approximation
- Penalized version for de-noising

- Computationally burdensome

Recovery Result for ℓ_1 Minimization

Theorem 3. [Donoho–Elad (2003), JAT (2003)] *Assume that* $\mathscr{D}$ *has quasi-coherence satisfying* $\mu_1(m-1) + \mu_1(m) < 1$*, and suppose that the vector* $\boldsymbol{x}$ *has an* exact representation *using* m *atoms. Then* ℓ_1 *minimization will recover this exact representation.*

Recovery Result for ℓ_1 Minimization

Theorem 3. [Donoho–Elad (2003), JAT (2003)] *Assume that $\mathscr{D}$ has quasi-coherence satisfying $\mu_1(m-1) + \mu_1(m) < 1$, and suppose that the vector $\boldsymbol{x}$ has an* exact representation *using m atoms. Then ℓ_1 minimization will recover this exact representation.*

Corollary 4. *Assume that $\mathscr{D}$ has coherence μ and that $m < \frac{1}{2}(\mu^{-1} + 1)$. If a vector $\boldsymbol{x}$ has an* exact representation *using m atoms, then ℓ_1 minimization will recover this exact representation.*

Recovery Result for ℓ_1 Minimization

Theorem 3. [Donoho–Elad (2003), JAT (2003)] *Assume that $\mathscr{D}$ has quasi-coherence satisfying $\mu_1(m-1) + \mu_1(m) < 1$, and suppose that the vector $\boldsymbol{x}$ has an* exact representation *using m atoms. Then ℓ_1 minimization will recover this exact representation.*

Corollary 4. *Assume that $\mathscr{D}$ has coherence μ and that $m < \frac{1}{2}(\mu^{-1} + 1)$. If a vector $\boldsymbol{x}$ has an* exact representation *using m atoms, then ℓ_1 minimization will recover this exact representation.*

- For the spike-sine dictionary, $m \leq \sqrt{d}/4$
- For the double-pulse dictionary, works for every m

State-of-Art for ℓ_1 Minimization

- Sharper conditions appear in [Fuchs 2003], [JAT 2003], [Gribonval-Nielsen 2003a, 2003b]
- These papers also study recovery of *exact representations*
- No general method is available for checking these general conditions

Natarajan's Result

- Back in 1995, Natarajan had already developed an approximation result for the forward selection algorithm
- The signal processing community is apparently unfamiliar with his work

Natarajan's Result

- Back in 1995, Natarajan had already developed an approximation result for the forward selection algorithm
- The signal processing community is apparently unfamiliar with his work
- His methods can be adapted to study OMP

Natarajan's Result

- Back in 1995, Natarajan had already developed an approximation result for the forward selection algorithm
- The signal processing community is apparently unfamiliar with his work
- His methods can be adapted to study OMP

Theorem 5. [Natarajan (1995), JAT (2003)] *Assume that* $\mathscr{D}$ *is a* non-redundant *dictionary, and suppose that it requires* m *terms to represent the vector* $\boldsymbol{x}$ *with tolerance* $\varepsilon/2$*. Then Orthogonal Matching Pursuit will compute a representation with error less than* ε *using no more than*

$$\frac{8m \, \ln(\|\boldsymbol{x}\|_2 / \varepsilon)}{\sigma_{\min}(\mathscr{D})} \qquad \textit{terms.}$$

- $Caveat\ lector:$ Natarajan's paper contains errors

Couvreur and Bresler's Result

- Couvreur and Bresler developed the first proof that an algorithm approximately solves the sparse problem over non-redundant dictionaries

Couvreur and Bresler's Result

- Couvreur and Bresler developed the first proof that an algorithm approximately solves the sparse problem over non-redundant dictionaries

Theorem 6. [Couvreur–Bresler (2000)] *Assume that* $\mathscr{D}$ *is a* non-redundant *dictionary. Suppose that the vector* $\boldsymbol{y}$ *has an exact representation using* m *terms. Then there is a number* $\delta > 0$ *so that* $\|\boldsymbol{x} - \boldsymbol{y}\|_2 < \delta$ *guarantees the backward elimination algorithm will recover the optimal* m*-term representation of* $\boldsymbol{x}$

Couvreur and Bresler's Result

- Couvreur and Bresler developed the first proof that an algorithm approximately solves the sparse problem over non-redundant dictionaries

Theorem 6. [Couvreur–Bresler (2000)] *Assume that* $\mathscr{D}$ *is a* non-redundant *dictionary. Suppose that the vector* $\boldsymbol{y}$ *has an exact representation using* m *terms. Then there is a number* $\delta > 0$ *so that* $\|\boldsymbol{x} - \boldsymbol{y}\|_2 < \delta$ *guarantees the backward elimination algorithm will recover the optimal* m*-term representation of* $\boldsymbol{x}$

- The algorithm recovers every vector with an exact representation

Couvreur and Bresler's Result

- Couvreur and Bresler developed the first proof that an algorithm approximately solves the sparse problem over non-redundant dictionaries

Theorem 6. [Couvreur–Bresler (2000)] *Assume that* $\mathscr{D}$ *is a* non-redundant *dictionary. Suppose that the vector* $\boldsymbol{y}$ *has an exact representation using* m *terms. Then there is a number* $\delta > 0$ *so that* $\|\boldsymbol{x} - \boldsymbol{y}\|_2 < \delta$ *guarantees the backward elimination algorithm will recover the optimal* m*-term representation of* $\boldsymbol{x}$

- The algorithm recovers every vector with an exact representation
- They provide no method for computing δ

Redundant Dictionaries, At Last

- In 2003, Gilbert, Muthukrishnan and Strauss published an efficient approximation algorithm for redundant dictionaries

Redundant Dictionaries, At Last

- In 2003, Gilbert, Muthukrishnan and Strauss published an efficient approximation algorithm for redundant dictionaries

Theorem 7. [GMS (2003)] *Assume that* $\mathscr{D}$ *has coherence* μ*, and let* $m < \frac{1}{8\sqrt{2}}\mu^{-1} - 1$*. For every vector* $\boldsymbol{x}$*, Orthogonal Matching Pursuit computes an* m*-term approximant* $\boldsymbol{a}_m$ *with error*

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \le 8\sqrt{m}\ \|\boldsymbol{x} - \boldsymbol{a}_{\mathrm{opt}}\|_2 .$$

Redundant Dictionaries, At Last

- In 2003, Gilbert, Muthukrishnan and Strauss published an efficient approximation algorithm for redundant dictionaries

Theorem 7. [GMS (2003)] *Assume that $\mathscr{D}$ has coherence μ, and let $m < \frac{1}{8\sqrt{2}}\,\mu^{-1} - 1$. For every vector $\boldsymbol{x}$, Orthogonal Matching Pursuit computes an m-term approximant $\boldsymbol{a}_m$ with error*

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq 8\sqrt{m}\ \|\boldsymbol{x} - \boldsymbol{a}_{\mathrm{opt}}\|_2\,.$$

Theorem 8. [GMS (2003)] *Assume that $\mathscr{D}$ has coherence μ, and let $m < \frac{1}{32}\,\mu^{-1}$. For every vector $\boldsymbol{x}$, the GMS algorithm computes an m-term approximant $\boldsymbol{a}_m$ with error*

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq \sqrt{1 + 2064\,\mu m^2}\ \|\boldsymbol{x} - \boldsymbol{a}_{\mathrm{opt}}\|_2\,.$$

Better Approximation with OMP

- JAT provided a new analysis of Orthogonal Matching Pursuit for quasi-incoherent dictionaries [2003]

Better Approximation with OMP

- JAT provided a new analysis of Orthogonal Matching Pursuit for quasi-incoherent dictionaries [2003]

Theorem 9. *Suppose that* $\mathscr{D}$ *has quasi-coherence* $\mu_1(m) < \frac{1}{2}$*. For an arbitrary signal* $\boldsymbol{x}$*, Orthogonal Matching Pursuit computes an* m*-term approximant* $\boldsymbol{a}_m$ *that satisfies*

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq \sqrt{1 + C(\mathscr{D}, m)}\ \|\boldsymbol{x} - \boldsymbol{a}_{\mathrm{opt}}\|_2 \,,$$

where we may estimate the constant as

$$C(\mathscr{D}, m) \leq \frac{m\,(1 - \mu_1(m))}{(1 - 2\,\mu_1(m))^2}.$$

Corollaries

Corollary 10. *Suppose that* $m < \frac{1}{2}\,\mu^{-1}$ *or (more generally) that* $\mu_1(m) < \frac{1}{2}$*. Then OMP recovers any signal that has an exact* m*-term representation.*

Corollaries

Corollary 10. *Suppose that* $m < \frac{1}{2}\mu^{-1}$ *or (more generally) that* $\mu_1(m) < \frac{1}{2}$. *Then OMP recovers any signal that has an exact* m*-term representation.*

Corollary 11. *Suppose that* $\mu_1(m) < \frac{1}{3}$. *For every signal* $\boldsymbol{x}$, *OMP computes an* m*-term approximant* $\boldsymbol{a}_m$ *that satisfies*

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq \sqrt{1 + 6m}\; \|\boldsymbol{x} - \boldsymbol{a}_{\text{opt}}\|_2 \,.$$

- For the spike-sine dictionary, this corollary applies whenever $m < \sqrt{d}/6$.
- For the double-pulse dictionary, this corollary applies for every m!

A New Algorithm

- The conference paper [TGMS 2003] presents a new greedy algorithm that achieves even better approximation bounds

A New Algorithm

- The conference paper [TGMS 2003] presents a new greedy algorithm that achieves even better approximation bounds

Theorem 12. *Suppose that $\mu_1(m) < \frac{1}{2}$. There is an algorithm that, for any vector $\boldsymbol{x}$, produces an m-term approximation $\boldsymbol{a}_m$ satisfying*

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq \sqrt{1 + C(\mathscr{D}, m)}\; \|\boldsymbol{x} - \boldsymbol{a}_{\text{opt}}\|_2\,,$$

We may bound the constant above using

$$C(\mathscr{D}, m) \leq \frac{2m\,\mu_1(m)}{(1 - 2\,\mu_1(m))^2}.$$

Corollaries

Corollary. *If* $\mu_1(m) \leq \min\{\frac{1}{4}, m^{-1}\}$*, the error bound simplifies to*

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq 3 \, \|\boldsymbol{x} - \boldsymbol{a}_{\mathrm{opt}}\|_2 .$$

Corollaries

Corollary. *If* $\mu_1(m) \leq \min\{\frac{1}{4}, m^{-1}\}$, *the error bound simplifies to*

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq 3 \, \|\boldsymbol{x} - \boldsymbol{a}_{\text{opt}}\|_2 \, .$$

- For the double-pulse dictionary, the theorem only provides error bound

$$\|\boldsymbol{x} - \boldsymbol{a}_m\|_2 \leq \sqrt{1 + 6m} \, \|\boldsymbol{x} - \boldsymbol{a}_{\text{opt}}\|_2$$

- Need $\mu_1(m) = O(m^{-1})$ to obtain significant savings

Overview of New Algorithm

A two-phase greedy pursuit:

- Use OMP to produce a partial approximation with moderate error
- Use Energy Pursuit to refine the first approximation

Energy Pursuit

- Fix a level of sparsity m
- Let $\boldsymbol{x}$ be a vector
- Select m atoms that carry the most energy:

$$\text{maximize} \quad \sum_{t=1}^{m} |\langle \boldsymbol{x}, \boldsymbol{\varphi}_{\lambda_t} \rangle|$$

Energy Pursuit

- Fix a level of sparsity m
- Let $\boldsymbol{x}$ be a vector
- Select m atoms that carry the most energy:

$$\text{maximize} \quad \sum_{t=1}^{m} |\langle \boldsymbol{x}, \boldsymbol{\varphi}_{\lambda_t} \rangle|$$

- For orthonormal bases, equivalent to truncation of Fourier expansion

Reference: [GMS 2003]

Combining the Phases

Suppose that an oracle provides the smallest number T so that T steps of Orthogonal Matching Pursuit yield an approximation $\boldsymbol{a}_T$ satisfying

$$\|\boldsymbol{x} - \boldsymbol{a}_T\|_2 \leq \sqrt{1 + \frac{m\,(1 - \mu_1(m))}{(1 - 2\,\mu_1(m))^2}}\; \|\boldsymbol{x} - \boldsymbol{a}_{\text{opt}}\|_2 \,.$$

Combining the Phases

Suppose that an oracle provides the smallest number T so that T steps of Orthogonal Matching Pursuit yield an approximation $\boldsymbol{a}_T$ satisfying

$$\|\boldsymbol{x} - \boldsymbol{a}_T\|_2 \leq \sqrt{1 + \frac{m\,(1 - \mu_1(m))}{(1 - 2\,\mu_1(m))^2}}\;\|\boldsymbol{x} - \boldsymbol{a}_{\text{opt}}\|_2\,.$$

The Algorithm

- Perform T steps of Orthogonal Matching Pursuit to get T atoms

Combining the Phases

Suppose that an oracle provides the smallest number T so that T steps of Orthogonal Matching Pursuit yield an approximation $\boldsymbol{a}_T$ satisfying

$$\|\boldsymbol{x} - \boldsymbol{a}_T\|_2 \leq \sqrt{1 + \frac{m\,(1 - \mu_1(m))}{(1 - 2\,\mu_1(m))^2}}\ \|\boldsymbol{x} - \boldsymbol{a}_{\mathrm{opt}}\|_2 \,.$$

The Algorithm

- Perform T steps of Orthogonal Matching Pursuit to get T atoms
- Perform Energy Pursuit on the residual to get $(m - T)$ more atoms

Combining the Phases

Suppose that an oracle provides the smallest number T so that T steps of Orthogonal Matching Pursuit yield an approximation $\boldsymbol{a}_T$ satisfying

$$\|\boldsymbol{x} - \boldsymbol{a}_T\|_2 \leq \sqrt{1 + \frac{m\,(1 - \mu_1(m))}{(1 - 2\,\mu_1(m))^2}}\ \|\boldsymbol{x} - \boldsymbol{a}_{\text{opt}}\|_2\,.$$

The Algorithm

- Perform T steps of Orthogonal Matching Pursuit to get T atoms
- Perform Energy Pursuit on the residual to get $(m - T)$ more atoms
- Compute the m-term approximation by projecting $\boldsymbol{x}$ onto the subspace spanned by the chosen atoms

Avoiding a Trip to Delphi

Method I

Guess the value of T by running the algorithm $(m+1)$ times with $T = 0, 1, \ldots, m$.

Avoiding a Trip to Delphi

Method I

Guess the value of T by running the algorithm $(m+1)$ times with $T = 0, 1, \ldots, m$.

Method II

Guess the optimal error by running the algorithm with errors taken from a geometric progression ranging between the machine precision and the norm of the input function.

Avoiding a Trip to Delphi

Method I

Guess the value of T by running the algorithm $(m+1)$ times with $T = 0, 1, \ldots, m$.

Method II

Guess the optimal error by running the algorithm with errors taken from a geometric progression ranging between the machine precision and the norm of the input function.

- Both methods are embarrassingly parallel, although efficient serial versions are also possible
- We can select the best of the multiple solutions

Approximate Nearest Neighbors

- Both phases of the algorithm require finding an atom from the dictionary that has maximal inner product with an input vector
- In a naïve implementation, this is the most time-consuming step

Approximate Nearest Neighbors

- Both phases of the algorithm require finding an atom from the dictionary that has maximal inner product with an input vector
- In a naïve implementation, this is the most time-consuming step
- We can quickly find inner products that are *nearly maximal* using an *Approximate Nearest Neighbors* data structure

Approximate Nearest Neighbors

- Both phases of the algorithm require finding an atom from the dictionary that has maximal inner product with an input vector
- In a naïve implementation, this is the most time-consuming step
- We can quickly find inner products that are *nearly maximal* using an *Approximate Nearest Neighbors* data structure
- The cost of a query is comparable to the cost of looking at each entry of the vector
- It takes significant preprocessing to build the data structure

Approximate Nearest Neighbors

- Both phases of the algorithm require finding an atom from the dictionary that has maximal inner product with an input vector
- In a naïve implementation, this is the most time-consuming step
- We can quickly find inner products that are *nearly maximal* using an *Approximate Nearest Neighbors* data structure
- The cost of a query is comparable to the cost of looking at each entry of the vector
- It takes significant preprocessing to build the data structure
- It can be shown that this implementation of the algorithm succeeds with slightly weaker error bounds

References: [Charikar 2003]

New Horizons

- Understand structured coherent dictionaries
- Develop approximation results for ℓ_1 minimization
- Study more sophisticated greedy algorithms
- Compute *a posteriori* error bounds
- Address subset selection problems
- Examine other sparsity measures
- Consider sparse approximation in Banach spaces
- Pursue simultaneous sparse approximation
- ...

Papers & Contact Information

- JAT. "Greed is Good: Algorithmic Results for Sparse Approximation." ICES Report 0304, The University of Texas at Austin, Feb. 2003.
- JAT. "Recovery of Short, Complex Linear Combinations via ℓ_1 Minimization." Unpublished note, Aug. 2003.
- TGMS. "Improved Sparse Approximation over Quasi-Incoherent Dictionaries." *Proc. of the 2003 Intl. Conf. on Image Processing*, Barcelona, Sept. 2003.
- Other material will appear in JAT's dissertation

- For more information, contact `<jtropp@ices.utexas.edu>`