Stronger Recovery Guarantees for Sparse Signals Exploiting Coherence Structure in Dictionaries

Eeshan Malhotra
Indian Institute of Technology Bombay

Karthik Gurumoorthy
International Centre for Theoretical Sciences, TIFR

Ajit Rajwade
Indian Institute of Technology Bombay

Introduction

- Signals that are sparse or compressible in some general basis (dictionary) can be recovered from significantly fewer measurements than the signal length (compressive sensing)
- Coherence-based recovery bounds for these exist, but are conservative
- Improved bounds characterize as a concatenation of two sub-parts
- We extend the result to arbitrary dictionaries, by devising the optimal artificially induced split in the dictionary
- A recursive algorithm also provides a heuristic to improve bounds even further, inducing a multi-way split in the dictionary
- The recovery algorithm is unchanged

Existing Foundational Results

- Consider a compressed measurement of a signal as
$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{\eta}$$
- In the special case where $\boldsymbol{D} = [\boldsymbol{A}\ \boldsymbol{B}]$ and $\boldsymbol{x}^{\boldsymbol{T}} = [\boldsymbol{x}_{\boldsymbol{a}}^{\boldsymbol{T}}\ \boldsymbol{x}_{\boldsymbol{b}}^{\boldsymbol{T}}]$, recovery guarantees can be improved as shown in [2]
- Define independent and mutual coherence values as:
$$\mu_a = \max_{i,j,i\neq j}\left|A_i^T A_j\right| \qquad \mu_b = \max_{i,j,i\neq j}\left|B_i^T B_j\right|$$
$$\mu_m = \max_{i,j}\left|A_i^T B_j\right| \qquad \mu_d = \max_{i,j,i\neq j}\left|D_i^T D_j\right|$$
- Studer and Baraniuk ([2]) showed that if
$$k < max\left\{\frac{2(1+\mu_a)}{\mu_a + 2\mu_d + \sqrt{\mu_a^2+\mu_m^2}},\ \frac{1+\mu_d}{2\mu_d}\right\}$$
- Then the solution $\hat{x}$ recovered using a basis pursuit approach is well bounded, and:
$$\left(1-\hat{\delta}\right)||x||_2^2 \ \leq\ ||Dx||_2^2 \ \leq\ \left(1+\hat{\delta}\right)||x||_2^2$$

With
$$\hat{\delta} = \min\left\{\frac{1}{2}\left(\mu_a(k-2) + k\sqrt{\mu_a^2+\mu_m^2}\right), \mu_d(k-1)\right\}$$

Key Ideas in Dictionary Splitting

- Extending the existing result to an arbitrary dictionary D involves devising an induced split. That is, Given a dictionary $\boldsymbol{D}$ devise matrices $\boldsymbol{A}$, B, such that $\boldsymbol{D}'=[\boldsymbol{A}\ \boldsymbol{B}]$, and the columns of $\boldsymbol{D}'$ can be permuted to give $\boldsymbol{D}$, so as to maximize F:
$$F = \frac{2(1+\mu_a)}{\mu_a + 2\mu_d + \sqrt{\mu_a^2+\mu_m^2}}$$
- We represent this problem as an equivalent graph theoretic problem
- Let G(V,E) be a fully connected graph such that each node in G corresponds to a column in D, and each edge weight is the dot product of the columns connected
- Now, the problem can be stated as: Define a cut through G, separating it into components $\boldsymbol{A}(\boldsymbol{V_a},\ \boldsymbol{E_a})$ and $\boldsymbol{B}(\boldsymbol{V_b},\ \boldsymbol{E_b})$ so as to maximize F, where
  - $\boldsymbol{\mu_a}$, $\boldsymbol{\mu_b}$: Heaviest edges in components $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively
  - $\boldsymbol{\mu_m}$: Heaviest edge crossing the cut

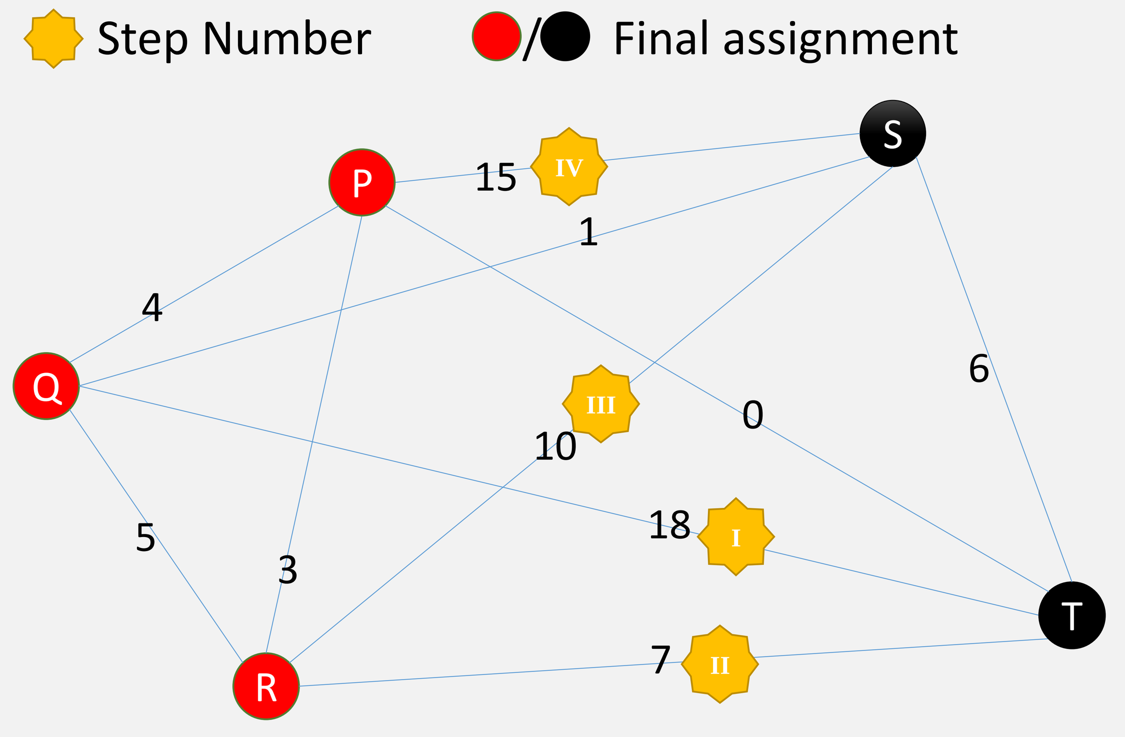
Algorithm - 2-Way Split - Intuition

- This is a greedy algorithm
- Start with the highest weight edge in the cut, assigning one node each to component A and B
- While not all vertices are assigned:
  - Candidate edges = edges with exactly one end-point assigned.
  - Pick max weight edge from candidates (say uv, with u assigned).
  - Assign v to the component that u is not in.
- There is a direct correspondence between columns of D and nodes of G. So D' is automatically recovered
- With some optimizations, the algorithm as formally described runs in time $O(n^2 \log n)$
- The algorithm is guaranteed to return the optimal 2-way split of the dictionary, as can be shown with an exchange argument (Refer to paper for proof)

Algorithm - 2-Way Split - Formal

```
for v ϵ V do:
        colour(v) ← 0
        L(v) ← edges from v, in decreasing order of weight
end for
E_x = cut edges ← ϕ
Q = list of edges to process ← ϕ
processed ← ϕ
e = uv ← heaviest edge in G
colour(u) ← 1
L(u) ← L(u) \ {e}
Q ← L(u)
processed ← u
while size(processed) < n do:
        if one of {u, v} (say v) is s.t. colour(v) == 0 then:
                if colour(u) == 1 then colour(v) ← 2
                if colour(u) == 2 then colour(v) ← 1
                L(v) ← L(v) \ {e}
                Merge L(v) into Q
                processed ← processed ∪ {v}
        end if
        if colour(u) != colour(v) then E_x ← E_x ∪ {e}
        e = uv ← Pop heaviest edge from Q
end while
V_a ← {v | colour(v) == 1}
V_b ← {v | colour(v) == 2}
Output V_a, V_b
```

Example Run



Algorithm - Multi-Way Split

- If the *effective* coherence of the dictionary is $\tilde{\mu}_d$
$$\tilde{\mu}_d(k-1) = \frac{1}{2}\left(\mu_a(k-2) + k\sqrt{\mu_a^2+\mu_m^2}\right)$$
- Or in the limiting case,
$$\tilde{\mu}_d = \frac{1}{2}\left(\mu_a + \sqrt{\mu_a^2+\mu_m^2}\right)$$
- But we could optimize μ_a in exactly the same way as we have used for μ_d . In fact, we could recursively apply this idea multiple times
- The multi-way split algorithm executed to depth l, achieves a 2^l-way split

```
function EFFECTIVE_COHERENCE(D, l)
        A, B = SPLIT2WAY(D)
        μ_a ← EFFECTIVE_COHERENCE(A, l-1)
        μ_b ← EFFECTIVE_COHERENCE(B, l-1)
        μ_m ← max_{i,j} {|A^T_i B^T_j |}
        μ_d ← max{μ_a, μ_b, μ_m}
        if l==1 then:
                return μ_d
        else
                return min { μ_d , 1/2 (μ_a + sqrt(μ_a^2 + μ_m^2)) }
end function

l ← maximum depth to explore
Output EFFECTIVE_COHERENCE(D, l)
```

- Heuristic for depth-l multi-way splitting
- Not guaranteed to produce the optimal split from the set of all possible 2^l-way splits
- The depth-l is a user-defined parameter that balances efficiency and computational time

Simulated Results

Simulated results, starting from a concatenation of two orthogonal dictionaries, with columns shuffled

	N	200	500	1000
Unsplit	μ_d	0.141	0.0894	0.0632
	k (Bound)	4.036	6.090	8.406
Split	μ_a	0	0	0
	μ_b	0	0	0
	μ_d	0.141	0.0894	0.0632
	$\tilde{\mu}_d$	0.0720	0.0448	0.0317
	k (Bound)	7.521	11.648	16.289
	$\hat{\delta}$	3.536	15.652	22.136

Conclusions

- While structure in dictionaries may not be naturally visible, such structure can be induced.
- The first algorithm presents a polynomial time algorithm to induce the optimal split to obtain the best tightest recovery guarantee, while the multi-way split extends it with a heuristic
- Bounds shown here may be looser than RIP bounds. However, computation of the RIC is computationally expensive. Our algorithm trades off some tightness for an efficiently computable bound
- Both proposed splitting algorithms only improve the theoretical bound on recovery for a generic dictionary, without altering the recovery algorithm

Key References

1. Candes, Romberg, Tao, *Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information*, IEEE Transactions on Information Theory, 2006.
2. Studer and Baraniuk, *Stable Restoration and Separation of Approximately Sparse Signals*, Applied and Computational Harmonic Analysis, 2014.