Learning With Structured Sparsity

Junzhou Huang, Tong Zhang, Dimitris Metaxas

Rutgers, The State University of New Jersey

Outline

- Motivation of structured sparsity
  - more priors improve the model selection stability
- Generalizing group sparsity: structured sparsity
  - CS: structured RIP requires fewer samples
  - statistical estimation: more robust to noise
  - examples of structured sparsity: graph sparsity
- An efficient algorithm for structured sparsity
  - StructOMP: structured greedy algorithm

Standard Sparsity

Suppose X the n × p data matrix. Let $Q(\mathbf{w}) = \|X\mathbf{w} - \mathbf{y}\|_2^2$.
The problem is formulated as

$$\min_{\mathbf{w}} Q(\mathbf{w}), \qquad \text{subject to } \|\mathbf{w}\|_0 \leq k$$

- Without priors for *supp(w)*
  - Convex relaxation (L1 regularization), such as Lasso
  - Greedy algorithm, such as OMP
- Complexity for k-sparse data O(k ln (p))
  - CS: related with the number of random projections
  - Statistics: related with the 2-norm estimation error

Group Sparsity

- Partition $\{1, \ldots, p\} = \cup_{j=1}^{m} G_j$ into m disjoint groups $G_1, G_2, \ldots, G_m$. Suppose *g* groups cover *k* features
- Priors for *supp(w)*
  - entries in one group are either zeros both or nonzeros both
- Group complexity: O(k + g ln(m)).
  - choosing *g* out of *m* groups (g ln(m)) for feature selection complexity (MDL)
  - suffer penalty k for estimation with *k* selected features (AIC)
  - Rigid, none-overlapping group setting

Motivation

- Dimension Effect
  - Knowing exact knowledge of *supp(w):* O(k) complexity
  - Lasso finds *supp(w)* with O(k ln(p)) complexity
  - Group Lasso finds *supp(w)* with O(g ln(m)) complexity
- Natural question
  - what if we have partial knowledge of *supp(w)*?
  - **structured sparsity**: not all feature combinations are equally likely, graph sparsity
  - complexity between k ln(p) and k.
  - More knowledge leads to the reduced complexity

Example

- Tree structured sparsity in wavelet compression
  - Original image
  - Recovery with unstructured sparsity, O(k ln p)
  - Recovery with structured sparsity, O(k)

Related Works (I)

- Bayesian framework for group/tree sparsity
  - Wipf&Rao 2007, Ji et al. 2008, He&Carin 2008
  - Empirical evidence and no theoretical results show how much better (under what kind of conditions)
- Group Lasso
  - Extensive literatures for empirical evidences (Yuan&Lin 2006)
  - Theoretical justifications (Bach 2008, Kowalski&Yuan 2008, Obozinski et al. 2008, Nardi&Rinaldo 2008, Huang&Zhang 2009)
  - Limitations: 1) inability for more general structure; 2) inability for overlapping groups

Related Works (II)

- Composite absolute penalty (CAP) [Zhao et al. 2006]
  - Handle overlapping groups; no theory for the effectiveness.
- Mixed norm penalty [Kowalski&Torresani 2009]
  - Structured shrinkage operations to identify the structure maps; no additional theoretical justifications
- Model based compressive sensing [Baraniuk et al. 2009]
  - Some theoretical results for the case in compressive sensing
  - No generic framework to flexibly describe a wide class of structures

Our Goal

- Empirical works evidently show better performance can be achieved with additional structures
- No general theoretical framework for structured sparsity that can quantify its effectiveness
- **Goals**
  - Quantifying structured sparsity;
  - Minimal number bounds of measurements required in CS;
  - estimation accuracy guarantee under stochastic noise;
  - A generic scheme and algorithm to flexible handle a wide class of structured sparsity problems

Structured Sparsity Regularization

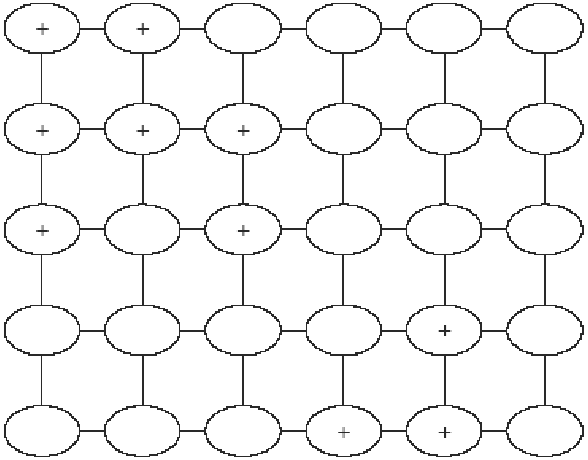
- Quantifying structure
  - *cl(F)*: number of binary bits to encode a feature set *F*;
  - Coding complexity: $s = c(F) = \underbrace{|F|}_{AIC} + \underbrace{cl(F)}_{MDL}$
  - number of samples needed in CS: $O(s)$
  - noise tolerance in learning is $O(s\sigma^2/n)$
- Assumption: not all sparse patterns are equally likely
- Optimization problem:

$$\min_{\mathbf{w}} Q(\mathbf{w}), \qquad \text{subject to } c(\text{supp}(\mathbf{w})) \leq s$$

Examples of structured sparsity

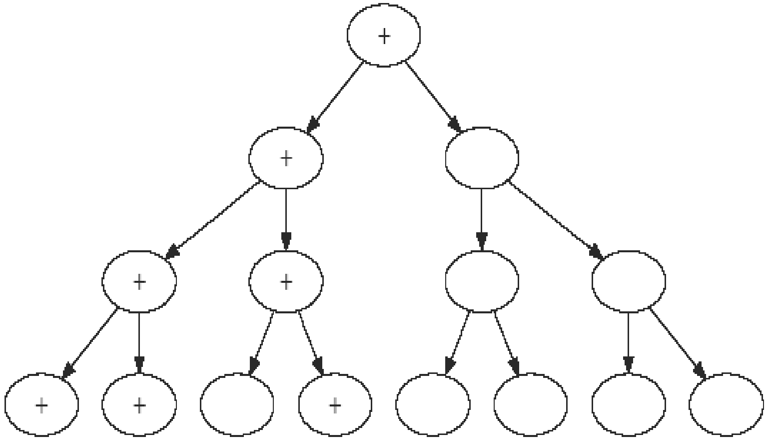
- Standard sparsity
  - complexity: s=O(k + k log(2p)) (k is sparsity number)
- Group sparsity: nonzeros tend to occur in groups
  - complexity: s=O(k + g log(2m))
- Graph sparsity (with O(1) maximum degree)
  - if a feature is nonzero, then near-by features are more likely to be nonzero. The complexity is s=O(k + g log p), where g is number of connected components.
- Random field sparsity:
  - any binary-random field probability distribution over the features induce a complexity as −log (probability).

Example: connected region



- A nonzero pixel implies adjacent pixels are more likely to be nonzeros
- The complexity is O(k + g ln p) where g is the number of connected components
- Practical complexity: O(k) with small g.

Example: hierarchical tree



- Parent nonzero implies children are more likely to be nonzeros.
- Complexity: O(k) instead of O(k ln p)
  - Requires parent as a feature if one child is a feature (zero-tree)
  - Implication: O(k) projections for wavelet CS

Proof Sketch of Graph Complexity

- Pick a starting point for every connected component
  - coding complexity is O(g ln p)
  - for tree, start from root with coding complexity 0
- Grow each feature node into adjacent nodes with coding complexity O(1)
  - require O(k) bits to code k nodes.
- Total is O(k + g ln p)

Solving Structured Sparsity

- Structured sparse eigenvalue condition: for n×p Gaussian projection matrix, any t > 0 and $\delta \in (0,1)$, let

$$n \geq \frac{8}{\delta^2}[\ln 3 + t + s\ln(1 + 8/\delta)] = O(s)$$

Then with probability at least $1 - e^{-t}$: for all vector $\mathbf{w} \in R^p$ with coding complexity no more than s:

$$(1-\delta)\|\mathbf{w}\|_2 \leq \frac{1}{\sqrt{n}}\|X\mathbf{w}\|_2 \leq (1+\delta)\|\mathbf{w}\|_2$$

Coding Complexity Regularization

- Coding complexity regularization formulation

$$OPT(s) = \min_{\mathbf{w}} Q(\mathbf{w}), \qquad \text{subject to } c(\mathrm{supp}(\mathbf{w})) \leq s$$

- With probability 1−η, the ε-OPT solution of coding complexity regularization satisfies:

$$\|X\hat{\mathbf{w}} - \mathbf{E}\mathbf{y}\|_2 \leq \inf_{c(\mathbf{w}) \leq s} \|X\mathbf{w} - \mathbf{E}\mathbf{y}\|_2 + \sigma\sqrt{2\ln(6/\eta)} + 2(7.4\sigma^2 s + 2.7\sigma^2 \ln(6/\eta) + \epsilon)^{1/2}$$

- Good theory but computationally inefficient.
  - convex relaxation: difficult to apply. In graph sparsity example, we need to search through connected components (dynamic groups) and penalize each group
  - Greedy algorithm, easy

StructOMP

- Repeat:
  - Find w to minimize Q(w) in the current feature set
  - select a block of features from a predefined “block set”, and add to the current feature set
- Block selection rule: compute the gain ratio:

$$\frac{Q(old) - Q(new)}{c(new) - c(old)},$$

and pick the feature-block to maximize the gain:

  - fastest objective value reduction per unit increase of coding complexity

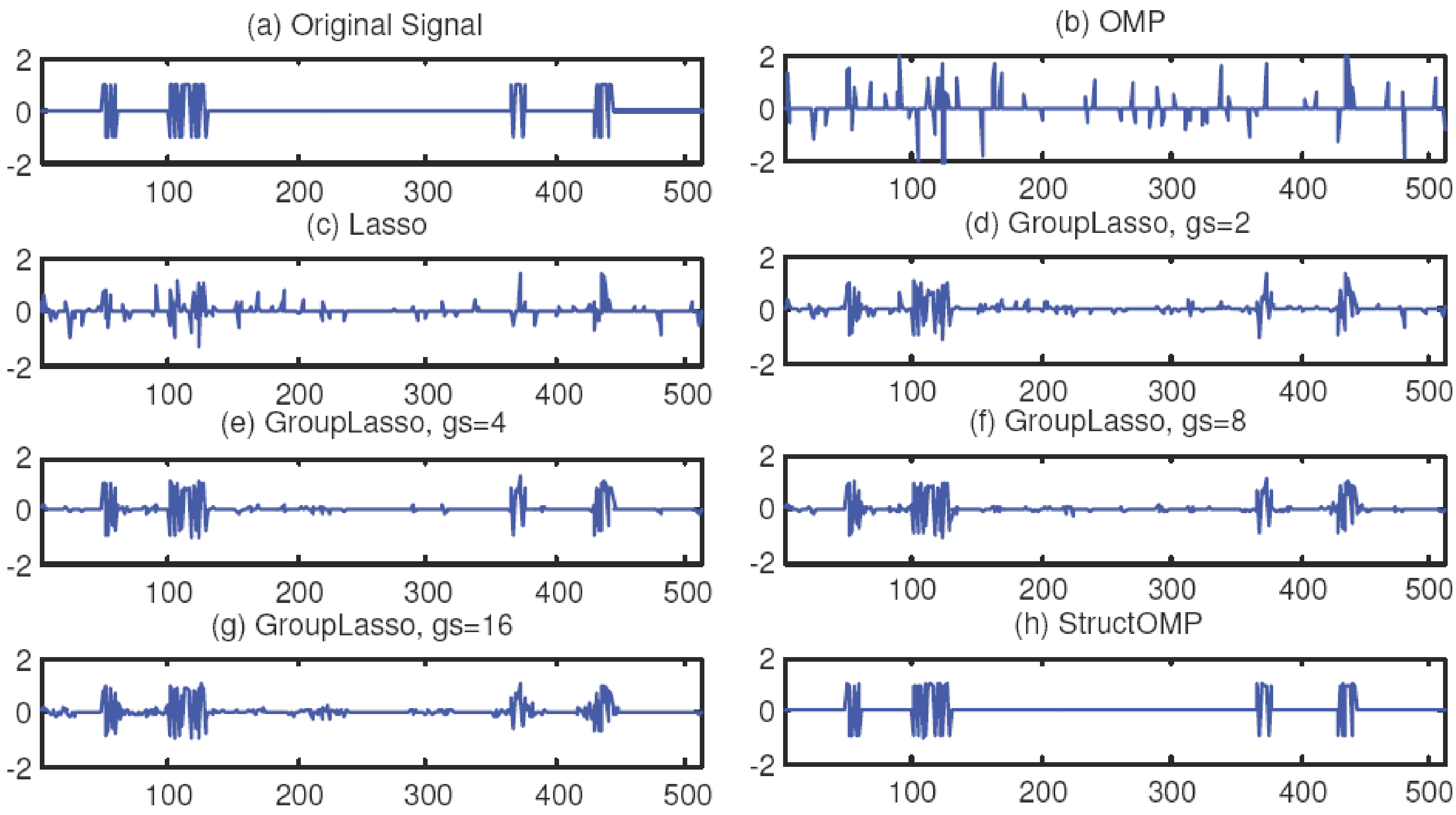
Convergence of StructOMP

- Assume structured sparse eigenvalue condition at each step
- StructOMP solution achieving OPT(s) +ε :
- Coding complexity regularization:
  - for strongly sparse signals (coefficients suddenly drop to zero; worst case scenario): solution complexity O(s log(1/ ε))
  - weakly sparse (coefficients decay to zero) q-compressible signals (decay at power q): solution complexity O(qs).

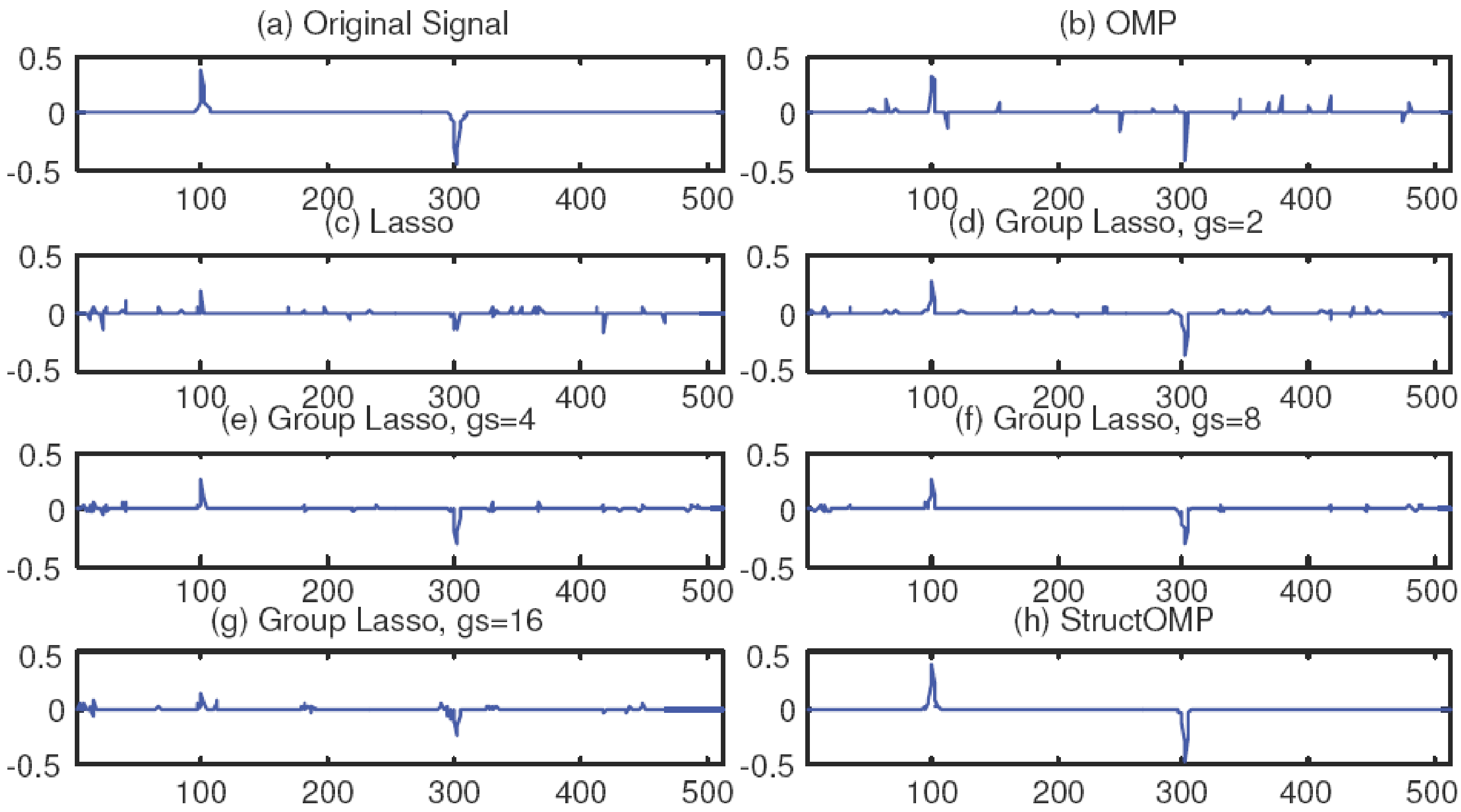
Experiments

- Focusing on graph sparsity
- Demonstrate the advantage of structured sparsity over standard/group sparsity. Compare the StructOMP with the OMP, Lasso and group Lasso
- The data matrix X are randomly generated with i.i.d draws from standard Gaussian distribution
- Quantitative evaluation: the recovery error is defined as the relative difference in 2-norm between the estimated sparse coefficient and the ground truth

Example: Strongly sparse signal



Example: Weakly sparse signal



Strong vs.Weak Sparsity

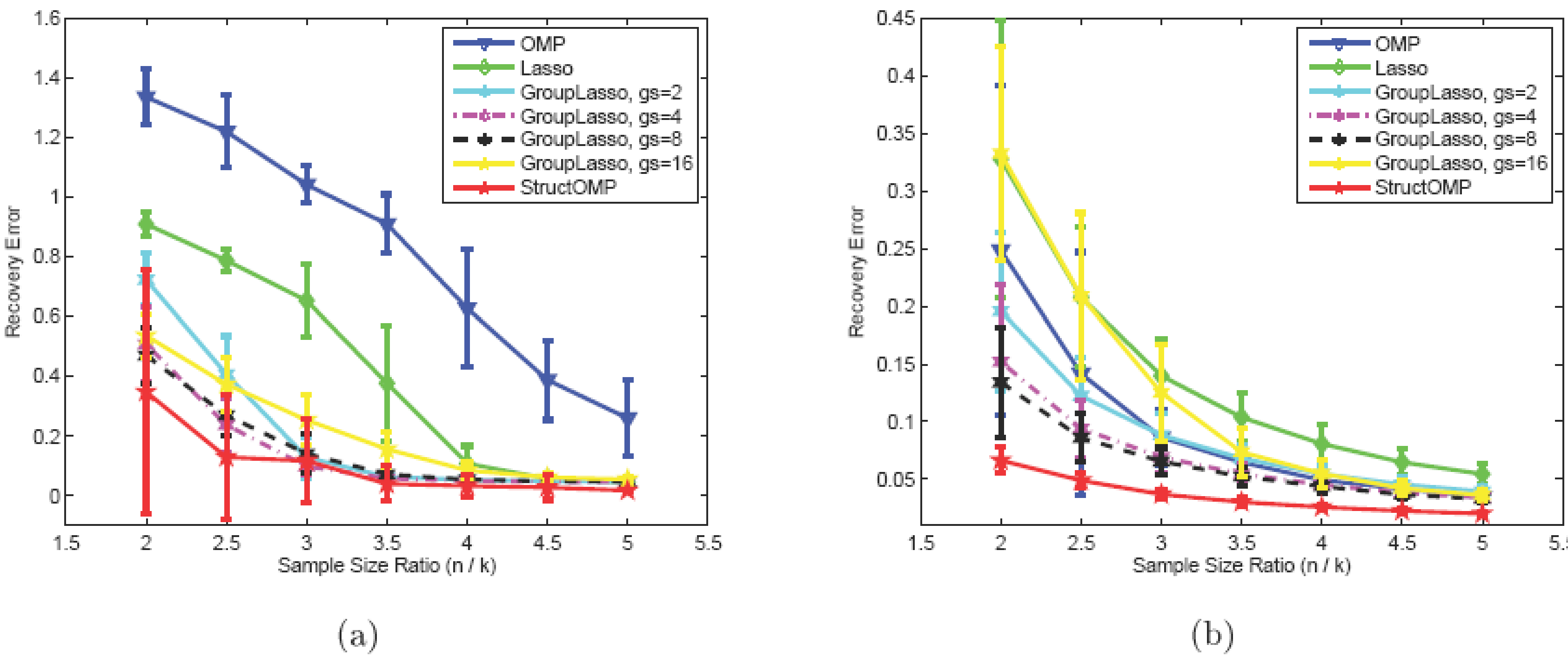
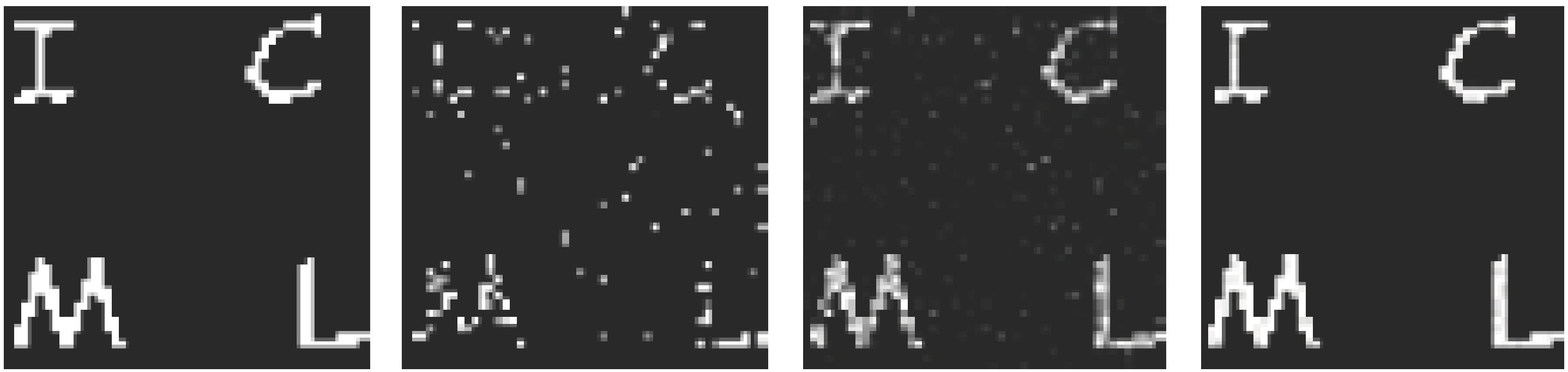


Figure. Recovery error vs. Sample size ratio (n/k): a) 1D strong sparse signals; (b) 1D Weak sparse signal

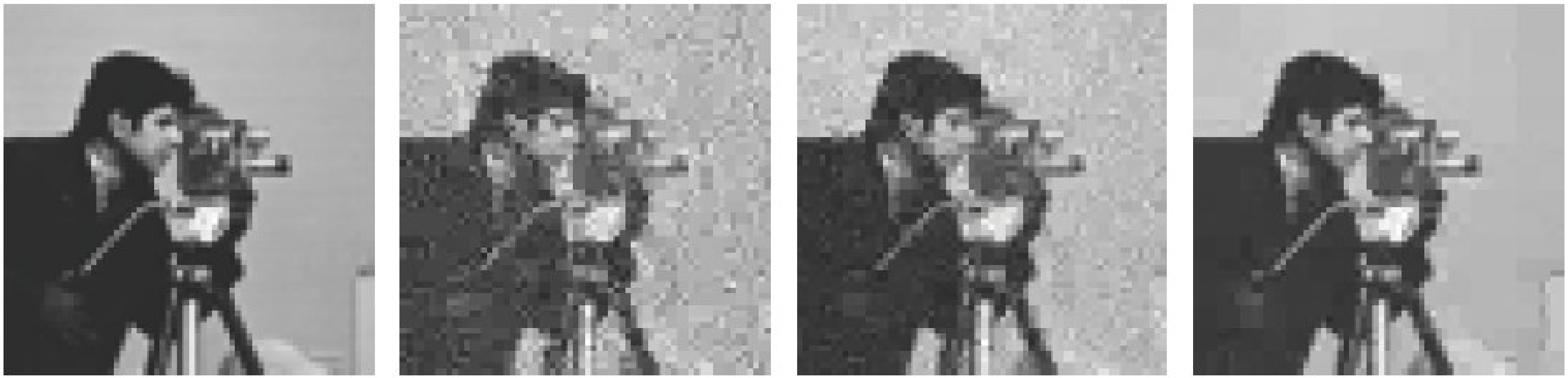
2D Image with Graph Sparsity



(a) (b) (c) (d)

Figure. Recovery results of a 2D gray image:
(a) original gray image, (b) recovered image with OMP (error is 0.9012),
(c) recovered image with Lasso (error is 0.4556) and (d) recovered image with StructOMP (error is 0.1528)

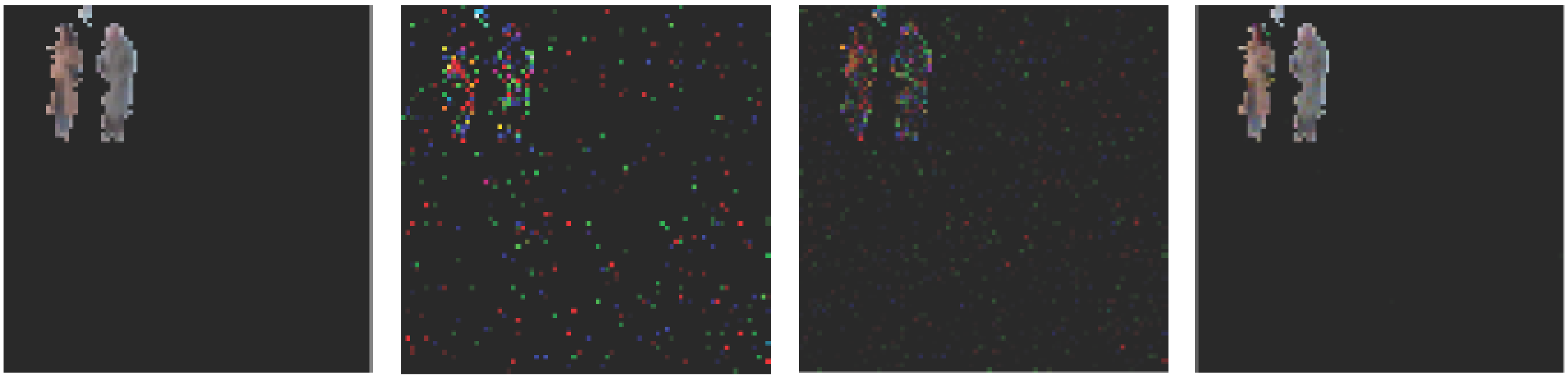
Hierarchical Structure in Wavelets



(a) (b) (c) (d)

Figure. Recovery results : (a) the original image, (b) recovered image with OMP (error is 0.21986), (c) recovered image with Lasso (error is 0.1670) and (d) recovered image with StructOMP (error is 0.0375)

Connected Region Structure



(a) (b) (c) (d)

Figure. Recovery results: (a) the background subtracted image, (b) recovered image with OMP (error is 1.1833), (c) recovered image with Lasso (error is 0.7075) and (d) recovered image with StructOMP (error is 0.1203)

Connected Region Structure

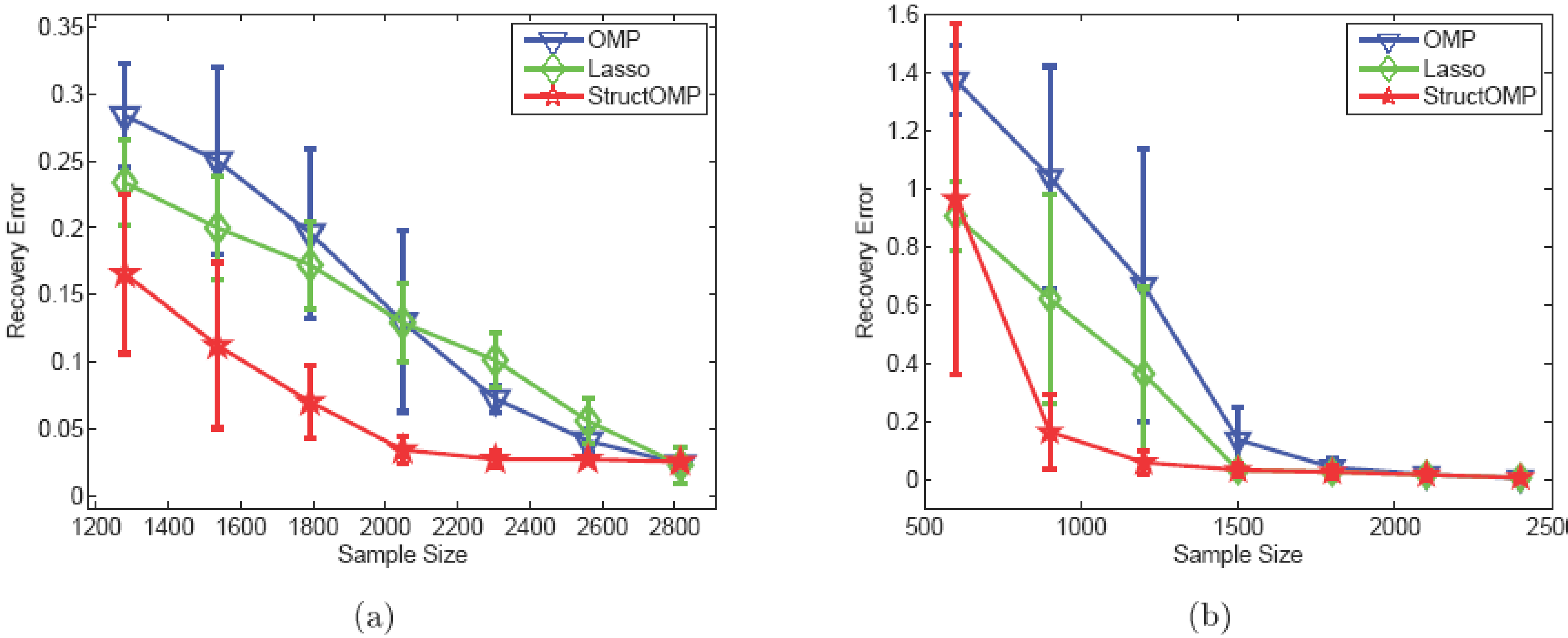


Figure. Recovery error vs. Sample size: a) 2D image with tree structured sparsity in wavelet basis; (b) background subtracted images with structured sparsity

Summary

- Proposed:
  - General theoretical framework for structured sparsity
  - Flexible coding scheme for structure descriptions
  - Efficient algorithm: StructOMP
  - Graph sparsity as examples
- Open questions
  - Backward steps
  - Convex relaxation for structured sparsity
  - More general structure representation

Thank you !