

# Compressed Sensing using Left and Right regular sparse graphs

**Avinash Vem**, Nagaraj Janakiraman, Krishna R. Narayanan

Department of Electrical and Computer Engineering  
Texas A&M University



# Outline

- 1 Introduction
  - Support Recovery
  - Known Limits
  - Main Result
  - Prior Work
- 2 Framework
  - Sensing Matrix
  - Decoding
- 3 Analysis
  - Peeling Decoder
  - Bin Decoder
- 4 Simulation Results

# Outline

- 1 Introduction
  - Support Recovery
  - Known Limits
  - Main Result
  - Prior Work
- 2 Framework
  - Sensing Matrix
  - Decoding
- 3 Analysis
  - Peeling Decoder
  - Bin Decoder
- 4 Simulation Results

# Compressed Sensing

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}$$

- $\mathbf{x}$  -  $N \times 1$  sparse signal
- $\text{supp}(\mathbf{x}) := \{i : x_i \neq 0, i \in [N]\}$
- $K = |\text{supp}(\mathbf{x})|$
- $\mathbf{A}$  -  $M \times N$  measurement matrix
- $\mathbf{w}$  - additive noise
- $\mathbf{y}$  -  $M \times 1$  measurement vector

## Sparsity

$$K \ll N$$

# Problem Statement

- Decoder: Given  $\mathbf{y}$  reconstruct the vector  $\mathbf{x}$  denoted by  $\hat{\mathbf{x}}$
- Prob. of failure of support recovery  $\mathbb{P}_F := \Pr(\text{supp}(\hat{\mathbf{x}}) \neq \text{supp}(\mathbf{x}))$
- Metrics of interest:
  - Sample complexity ( $M$ )
  - Decoding complexity
  - $\mathbb{P}_F$

# Problem Statement

- Decoder: Given  $\mathbf{y}$  reconstruct the vector  $\mathbf{x}$  denoted by  $\hat{\mathbf{x}}$
- Prob. of failure of support recovery  $\mathbb{P}_F := \Pr(\text{supp}(\hat{\mathbf{x}}) \neq \text{supp}(\mathbf{x}))$
- Metrics of interest:
  - Sample complexity ( $M$ )
  - Decoding complexity
  - $\mathbb{P}_F$

## Objective

Devise a scheme with minimal num. of measurements  $M$  and minimal decoding complexity such that  $\mathbb{P}_F \rightarrow 0$  as  $N(\text{and } K) \rightarrow \infty$

## Optimal order for Support Recovery [1]

- In the sub-linear sparsity regime,  $K = o(N)$ , necessary and sufficient conditions are shown to be:

$$C_1 K \log \left( \frac{N}{K} \right) < M < C_2 K \log \left( \frac{N}{K} \right)$$

- In the linear sparsity regime,  $K = \alpha N$ , it was shown that  $M = \Theta(N)$  measurements are sufficient for asymptotically reliable recovery.

## Optimal order for Support Recovery [1]

- In the sub-linear sparsity regime,  $K = o(N)$ , necessary and sufficient conditions are shown to be:

$$C_1 K \log \left( \frac{N}{K} \right) < M < C_2 K \log \left( \frac{N}{K} \right)$$

- In the linear sparsity regime,  $K = \alpha N$ , it was shown that  $M = \Theta(N)$  measurements are sufficient for asymptotically reliable recovery.

- In [1], the minimum value of the signal space affects the bounds on  $M$

$$x_i \in \mathcal{X} \triangleq \{Ae^{i\theta} : A \in \mathcal{A}, \theta \in \Omega\} \cup \{0\},$$

$$\mathcal{A} = \{A_{\min} + \rho l\}_{l=0}^{L_1}, \Omega = \{2\pi l/L_2\}_{l=0}^{L_2}$$

[1] Information Theoretic Limits of Support Recovery- Wainwright-2007



# Main result

## Optimal Sample and Decoding Complexities

In the sub-linear sparsity regime, for a given SNR of  $\frac{A_{\min}^2}{\sigma^2}$ , our scheme has

- Sample complexity of  $M = c_1 K \log(\frac{c_2 N}{K})$
- Decoding complexity of  $O(K \log(\frac{N}{K}))$
- $\mathbb{P}_F \rightarrow 0$  asymptotically in  $K$

where the constants  $c_1$  and  $c_2$  are dependent on SNR, desired rate of decay of  $\mathbb{P}_F$  and left degree  $\ell$ .

# Main result

## Optimal Sample and Decoding Complexities

In the sub-linear sparsity regime, for a given SNR of  $\frac{A_{\min}^2}{\sigma^2}$ , our scheme has

- Sample complexity of  $M = c_1 K \log(\frac{c_2 N}{K})$
- Decoding complexity of  $O(K \log(\frac{N}{K}))$
- $\mathbb{P}_F \rightarrow 0$  asymptotically in  $K$

where the constants  $c_1$  and  $c_2$  are dependent on SNR, desired rate of decay of  $\mathbb{P}_F$  and left degree  $\ell$ .

## Linear Sparsity Regime

In the linear sparsity regime our scheme has

- Sample complexity of  $M = c_3 K \log(K)$
- Decoding complexity of  $O(K \log(K))$
- $\mathbb{P}_F \rightarrow 0$  asymptotically in  $K$

where the constant  $c_3 > 1$  is a parameter dependent on left degree  $\ell$ .

- Zhang and Pfister, “Verification Decoding of High-Rate LDPC Codes With Applications in Compressed Sensing”, 2008
- Jafarpour, Xu, Hassibi and Calderbank, “Efficient and robust compressed sensing using optimized expander graphs”, 2009
  - Sample complexity of  $O(K \log N)$  and decoding complexity of  $O(K)$  for noiseless setting

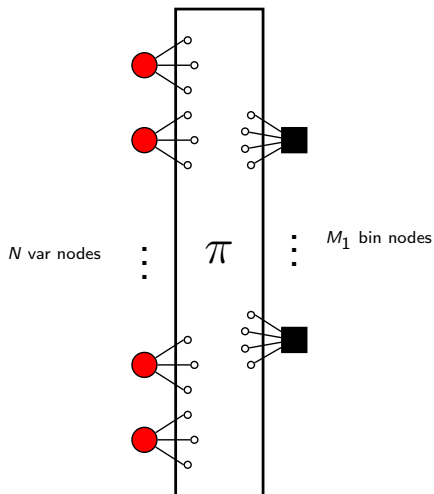
- Zhang and Pfister, “Verification Decoding of High-Rate LDPC Codes With Applications in Compressed Sensing”, 2008
- Jafarpour, Xu, Hassibi and Calderbank, “Efficient and robust compressed sensing using optimized expander graphs”, 2009
  - Sample complexity of  $O(K \log N)$  and decoding complexity of  $O(K)$  for noiseless setting
- Li, Pedarsani and Ramchandran, “Sub-linear compressed sensing for support recovery using sparse-graph codes”, 2014
  - Introduced sparse-graph codes peeling decoder framework to CS
  - Sample and measurement complexities of  $O(K \log N)$  for noisy setting
  - Sample and measurement complexities of  $2K$  and  $O(K)$  for noiseless setting

# Outline

- 1 Introduction
  - Support Recovery
  - Known Limits
  - Main Result
  - Prior Work
- 2 Framework
  - Sensing Matrix
  - Decoding
- 3 Analysis
  - Peeling Decoder
  - Bin Decoder
- 4 Simulation Results

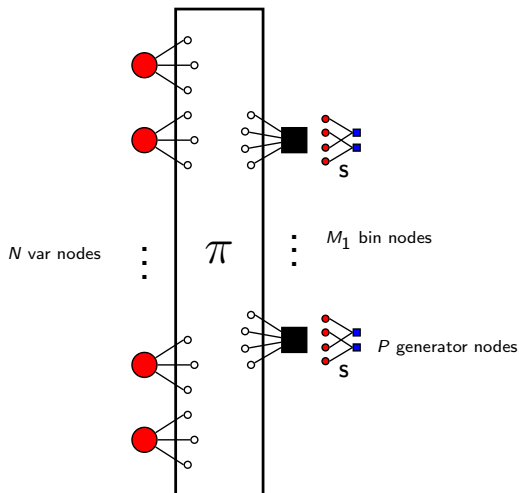
# Graphical Representation

$(N, \ell, r, \mathbf{S})$  ensemble.  $\ell N = rM_1$ .



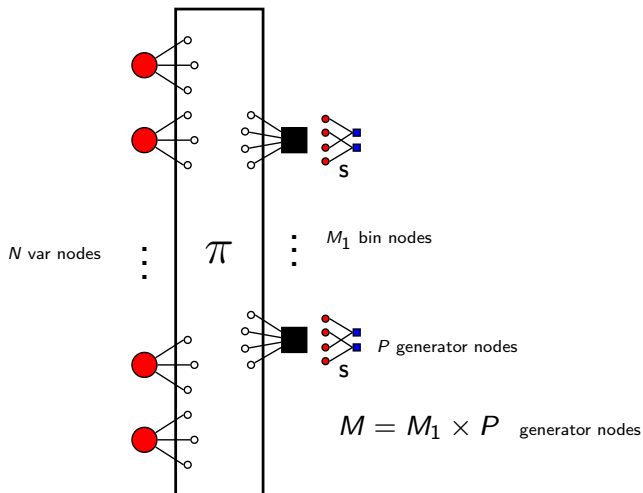
# Graphical Representation

$(N, \ell, r, \mathbf{S})$  ensemble.  $\ell N = rM_1$ .  $\dim(\mathbf{S}) = P \times r$ .



# Graphical Representation

$(N, \ell, r, \mathbf{S})$  ensemble.  $\ell N = rM_1$ .  $\dim(\mathbf{S}) = P \times r$ .





# Matrix Representation

$(N, \ell, r, \mathbf{S})$  ensemble.

- $\mathbf{H}$  be the adjacency matrix (binning operation)-  $M_1 \times N$
- $\mathbf{S}$  be the generator matrix at each bin -  $P \times r$

$$\tilde{\mathbf{y}} = \mathbf{H}(\mathbf{x}) = \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{y}}_2 \\ \vdots \\ \tilde{\mathbf{y}}_{M_1} \end{bmatrix}, \dim(\tilde{\mathbf{y}}_i) = r \times 1,$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_{M_1} \end{bmatrix}, \text{ where } \mathbf{y}_i = \mathbf{S}\tilde{\mathbf{y}}_i, \dim(\mathbf{y}_i) = P \times 1$$

- We define a tensor operation such that

$$\mathbf{y} = (\mathbf{S} \boxplus \mathbf{H})\mathbf{x}$$

# Tensor Operation

- Sensing matrix  $\mathbf{A}_{M_1 P \times N} = S_{P \times r} \boxplus H_{M_1 \times N}$  where

# Tensor Operation

- Sensing matrix  $\mathbf{A}_{M_1 P \times N} = \mathbf{S}_{P \times r} \boxplus \mathbf{H}_{M_1 \times N}$  where
- $\forall i \in [1 : M_1]$ , define a  $P \times N$  matrix

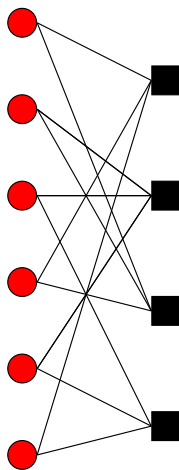
$$\mathbf{S}_i = \mathbf{h}_i \boxtimes \mathbf{S} \triangleq [\mathbf{0}, \dots, \mathbf{0}, \mathbf{s}_1, \mathbf{0}, \dots, \mathbf{s}_2, \dots, \mathbf{0}, \mathbf{s}_r, \mathbf{0}]$$

where the  $r$  columns are placed in the  $r$  non-zero indices of  $\mathbf{h}_i$ .

- $\mathbf{S} \boxplus \mathbf{H} = \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \vdots \\ \mathbf{S}_{M_1} \end{bmatrix}$

# Example

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

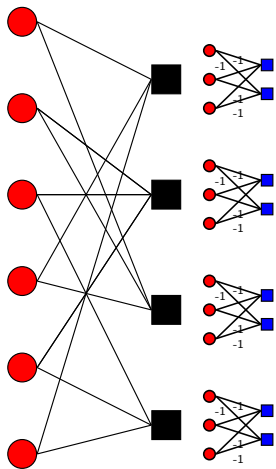


# Example

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

and

$$\mathbf{S} = \begin{bmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \end{bmatrix}.$$



# Example

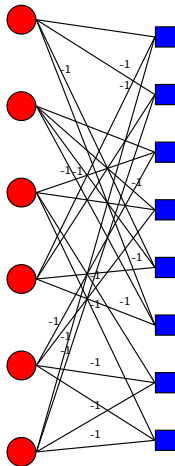
$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

and

$$\mathbf{S} = \begin{bmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \end{bmatrix}.$$

Sensing matrix  $\mathbf{A}$  with  $M = 8$ :

$$\mathbf{A} = \mathbf{H} \boxplus \mathbf{S} = \begin{bmatrix} +1 & 0 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & +1 & 0 & -1 \\ 0 & +1 & -1 & 0 & -1 & 0 \\ 0 & -1 & +1 & 0 & -1 & 0 \\ +1 & -1 & 0 & -1 & 0 & 0 \\ -1 & +1 & 0 & -1 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 & -1 \\ 0 & 0 & -1 & 0 & +1 & -1 \end{bmatrix}$$



# Bin Decoding

At each bin, input to the decoder is

$$\mathbf{y}_i = \sum_{j=1}^r x_{\mathbf{h}_i^j} \mathbf{s}_j + \mathbf{w}_i$$

- Zero-ton: Is it just noise?

$$\hat{\mathcal{H}}_i = \mathcal{H}_Z, \quad \text{if } \frac{1}{P} \|\mathbf{y}_i\|^2 \leq (1 + \gamma) \sigma^2$$

- Singleton: If a single variable is non-zero?

$$\alpha_k = \frac{\mathbf{s}_k^\dagger \mathbf{y}_i}{\|\mathbf{s}_k\|^2}$$

$$\hat{k} = \arg \min_k \|\mathbf{y}_i - \alpha_k \mathbf{s}_k\|$$

$$\hat{x}[\hat{k}] = \arg \min_{x \in \mathcal{X}} \|x - \alpha_{\hat{k}}\|$$

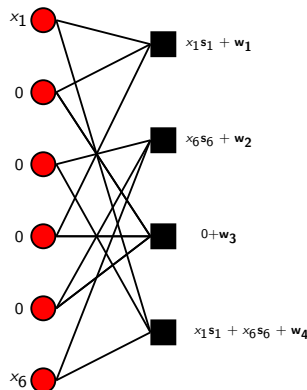
- Multi-ton: More than one non-zero variable?

$$\hat{\mathcal{H}}_i = \mathcal{H}_S(\hat{k}, \hat{x}[\hat{k}]), \quad \text{if } \frac{1}{P} \|\mathbf{y}_i - \hat{x}[\hat{k}] \mathbf{s}_{\hat{k}}\|^2 \leq (1 + \gamma) \sigma^2$$

# Peeling Decoding

```

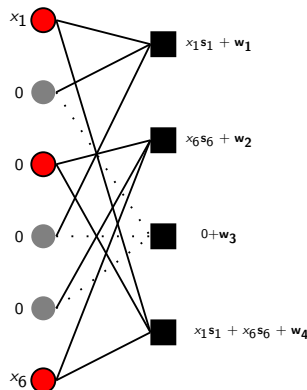
while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z$  or  $\mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
    connected
  
```





# Peeling Decoding

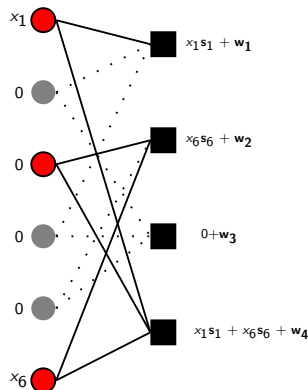
**while**  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z$  or  $\mathcal{H}_S$ , **do**  
     **if**  $\mathcal{H}_i = \mathcal{H}_Z$  **then**  
         Remove the bin  $i$   
         Assign 0 to all the variables connected  
     **else if**  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  **then**  
         Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$   
         Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$   
         Remove the bin and all variables  
         connected



# Peeling Decoding

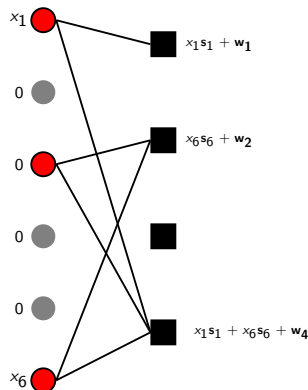
```

while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z$  or  $\mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
    connected
  
```



# Peeling Decoding

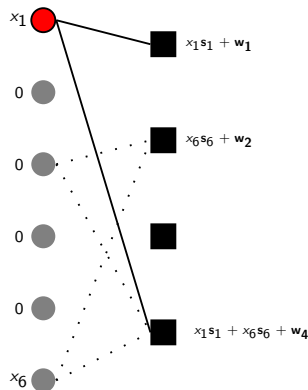
**while**  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z$  or  $\mathcal{H}_S$ , **do**  
     **if**  $\mathcal{H}_i = \mathcal{H}_Z$  **then**  
         Remove the bin  $i$   
         Assign 0 to all the variables connected  
     **else if**  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  **then**  
         Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$   
         Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$   
         Remove the bin and all variables  
         connected



# Peeling Decoding

```

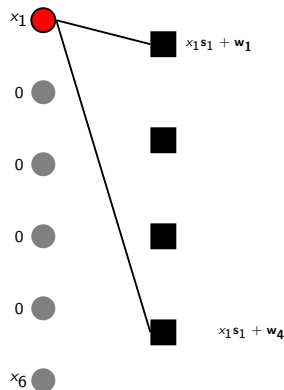
while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z \text{ or } \mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
    connected
  
```



# Peeling Decoding

```

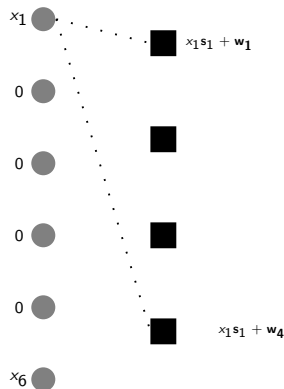
while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z \text{ or } \mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
    connected
  
```



# Peeling Decoding

```

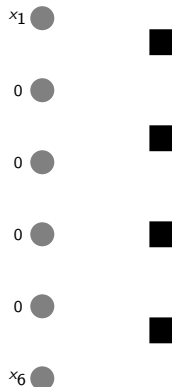
while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z \text{ or } \mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
connected
  
```



# Peeling Decoding

```

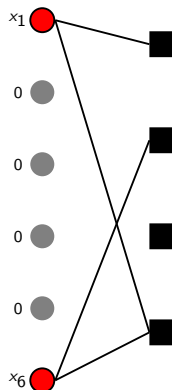
while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z$  or  $\mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
    connected
  
```



# Peeling Decoding

```

while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z \text{ or } \mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
    connected
  
```

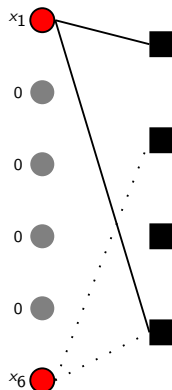




# Peeling Decoding

```

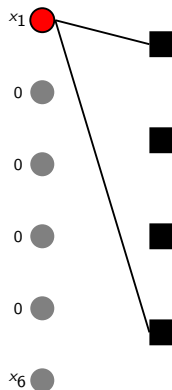
while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z \text{ or } \mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
    connected
  
```



# Peeling Decoding

```

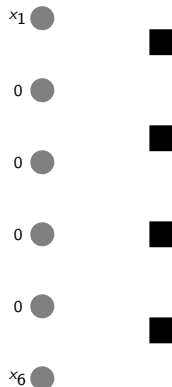
while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z \text{ or } \mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
    connected
  
```



# Peeling Decoding

```

while  $\exists i \in [M_1] : \mathcal{H}_i = \mathcal{H}_Z$  or  $\mathcal{H}_S$ , do
  if  $\mathcal{H}_i = \mathcal{H}_Z$  then
    Remove the bin  $i$ 
    Assign 0 to all the variables connected
  else if  $\mathcal{H}_i = \mathcal{H}_S(k, x[k])$  then
    Assign  $x[k]$  to  $k^{\text{th}}$  variable in bin  $i$ 
    Subtract  $x[k]\mathbf{s}_k$  from connected  $\mathbf{y}_i$ 
    Remove the bin and all variables
    connected
  
```



# Outline

- 1 Introduction
  - Support Recovery
  - Known Limits
  - Main Result
  - Prior Work
- 2 Framework
  - Sensing Matrix
  - Decoding
- 3 Analysis
  - Peeling Decoder
  - Bin Decoder
- 4 Simulation Results

# Oracle based Peeling Decoder

- Assume the hypothesis detection in each bin is correct
- Equivalence to peeling decoder on pruned graph- all zero variables are removed

Equivalence to  $(N, l, r)$  LDPC on  $\text{BEC}(\epsilon = \frac{\kappa}{N})$

If  $\text{supp}(\mathbf{x}) = \{i : y_i = \mathcal{E}\}$ , then  $P_{\text{BEC}}^{(i)}(\mathbf{y}) = P_{\text{SR}}^{(i)}(\mathbf{z})$  for  $\mathbf{z} = \mathbf{H}\mathbf{x}$ .

# Oracle based Peeling Decoder

- Assume the hypothesis detection in each bin is correct
- Equivalence to peeling decoder on pruned graph- all zero variables are removed

Equivalence to  $(N, l, r)$  LDPC on  $\text{BEC}(\epsilon = \frac{K}{N})$

If  $\text{supp}(\mathbf{x}) = \{i : y_i = \mathcal{E}\}$ , then  $P_{\text{BEC}}^{(i)}(\mathbf{y}) = P_{\text{SR}}^{(i)}(\mathbf{z})$  for  $\mathbf{z} = \mathbf{H}\mathbf{x}$ .

- Choose  $M_1 = \eta K$  thus  $r = \frac{\ell N}{\eta K}$

DE for Peeling decoder on LDPC -BEC channel

Fractional number of degree one checks remaining

$$\tilde{R}_1(y) = r\epsilon y^{l-1}[y - 1 + (1 - \epsilon y^{l-1})^{r-1}]$$

where  $\epsilon = \frac{K}{N}$  and  $r = \frac{\ell N}{\eta K}$

## Peeling threshold

$\eta^{\text{Th}}$  is defined to be the minimum value of  $\eta$  for which there is no non-zero solution for the equation:

$$\begin{aligned} y &= \lim_{\frac{N}{K} \rightarrow \infty} 1 - \left( 1 - \frac{Ky^{\ell-1}}{N} \right)^{\frac{\ell N}{\eta K}} \\ &= 1 - e^{\frac{-\ell y^{\ell-1}}{\eta}} \end{aligned}$$

in the range  $y \in [0, 1]$ .

## Peeling threshold

$\eta^{\text{Th}}$  is defined to be the minimum value of  $\eta$  for which there is no non-zero solution for the equation:

$$\begin{aligned} y &= \lim_{\frac{N}{K} \rightarrow \infty} 1 - \left( 1 - \frac{Ky^{\ell-1}}{N} \right)^{\frac{\ell N}{\eta K}} \\ &= 1 - e^{\frac{-\ell y^{\ell-1}}{\eta}} \end{aligned}$$

in the range  $y \in [0, 1]$ .

## Threshold behavior

For  $M_1 > \eta^{\text{Th}} K$  bin nodes, the peeling decoder will be successful with probability  $1 - O\left(\frac{1}{K^{\ell-2}}\right)$

Note that  $\eta^{\text{Th}}$  is a function of just the left degree  $\ell$ .



# Bin detection matrix

- Singleton detection is the crucial part of bin decoding:

$$\mathbf{y}_i = x_k \mathbf{s}_k + \mathbf{w}_i$$

- Error correction coding:  $\mathbf{S}$  be the codebook, where each  $\mathbf{s}_i$  is a codeword.
- Block length  $= P$ . # codewords  $\geq \frac{N\ell}{\eta K}$
- Choose a code with rate  $R(\beta)$  s.t. fractional minimum distance

$$\beta > \mathbb{P}_e := e^{-\frac{A_{\min}^2}{2\sigma^2}}$$

- Thus  $P = \frac{\lceil \log_2(\frac{N\ell}{\eta K}) \rceil}{R(\beta)}$ .

## Sample Complexity

$$\begin{aligned} M &= M_1 \times P \\ &\geq \left\lceil \frac{\eta^{\text{Th}}}{R(\mathbb{P}_e)} \right\rceil K \log \left( \frac{\ell N}{\eta^{\text{Th}} K} \right) \end{aligned}$$

# Analysis of Bin Decoding

- Let  $E_{\text{bin}}$  be the event an error was made in overall bin decoding
- Union bounding:  $E_{\text{bin}} \leq (\eta K + \ell K) \Pr(E)$

Error Probability of a bin [LPR-2014]

$$\Pr(E) \leq 3e^{-\frac{P}{4} \frac{\gamma^2}{1+4\gamma}} + 2e^{-\frac{P}{4} (\sqrt{1+2\gamma}-1)^2} + 4e^{-c_6 P \left(1 - \frac{2\gamma\sigma^2}{A_{\min}^2}\right)} + 2e^{-P \frac{(\beta - \mathbb{P}_e)^2}{2\mathbb{P}_e(1-\mathbb{P}_e)}}$$

# Analysis of Bin Decoding

- Let  $E_{\text{bin}}$  be the event an error was made in overall bin decoding
- Union bounding:  $E_{\text{bin}} \leq (\eta K + \ell K) \Pr(E)$

Error Probability of a bin [LPR-2014]

$$\Pr(E) \leq 3e^{-\frac{P}{4} \frac{\gamma^2}{1+4\gamma}} + 2e^{-\frac{P}{4} (\sqrt{1+2\gamma}-1)^2} + 4e^{-c_6 P \left(1 - \frac{2\gamma\sigma^2}{A_{\min}^2}\right)} + 2e^{-P \frac{(\beta - \mathbb{P}_e)^2}{2\mathbb{P}_e(1-\mathbb{P}_e)}}$$

## Sub-Linear sparsity

- Order optimal sample complexity with precise constants given
- $\mathbb{P}_F \rightarrow 0$  as  $N$  (and  $K$ )  $\rightarrow \infty$
- Trade-off between the constants in  $M$ , rate of decay of  $\mathbb{P}_F$  and SNR
- Optimal decoding complexity of  $O\left(K \log\left(\frac{N}{K}\right)\right)$

# Implications

Error Probability of a bin - Ramchandran *et al*, 2014

$$\Pr(E) \leq 3e^{-\frac{P}{4} \frac{\gamma^2}{1+4\gamma}} + 2e^{-\frac{P}{4}(\sqrt{1+2\gamma}-1)^2} + 4e^{-c_6 P \left(1 - \frac{2\gamma\sigma^2}{A_{\min}^2}\right)} + 2e^{-P \frac{(\beta - \mathbb{P}_e)^2}{2\mathbb{P}_e(1-\mathbb{P}_e)}}$$

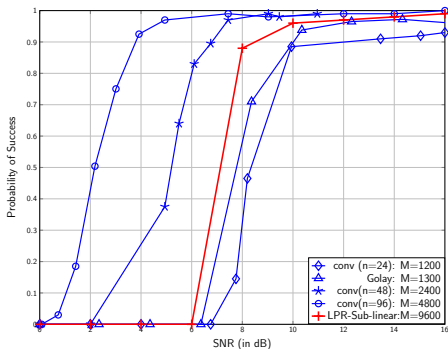
**Linear sparsity:**  $K = \alpha N$

- Choice of  $P = c_1 \log \left( c_2 \frac{N}{K} \right)$  doesn't work
- We choose  $P = \log(K)$  and rate  $R(\beta)$  as earlier
- A sub-code of size  $\frac{\ell}{\alpha\eta}$  of the codebook is chosen as  $\mathbf{S}$
- Sample complexity of  $\eta^{\text{Th}} K \log K$
- Can we do  $\Theta(K)$  with practical decoding?

# Outline

- 1 Introduction
  - Support Recovery
  - Known Limits
  - Main Result
  - Prior Work
- 2 Framework
  - Sensing Matrix
  - Decoding
- 3 Analysis
  - Peeling Decoder
  - Bin Decoder
- 4 Simulation Results

- $K = 50, N = 10^5$ .  $\mathcal{X} = \{+1, -1\}$
- $\ell = 4, \eta = 2(M_1 = 2K)$ . Note that  $\eta^{\text{Th}}(\ell = 4) = 1.295$
- $r = \frac{N\ell}{\eta K} = 4000$ .  $\log_2(r) = 12$
- (12, 24) Golay code and (12,  $n$ ) convolutional codes for  $n = 24, 48, 96$  with QAM form **S**.



# Questions?