LDPC Codes for Dictionary Learning

10701 Project

Haewon Jeong, Yongjune Kim and
Mark Whiting

Background

- **Compressed sensing & Dictionary learning**

$$\min \ \|\alpha\|_0$$
$$s.t. \quad D\alpha = \mathbf{x}$$

sparse signal measurement

  - Measurement matrix (dictionary) $D \in \mathcal{R}^{m \times k}$
  - $\boldsymbol{\alpha} \in \mathcal{R}^k, \mathbf{x} \in \mathcal{R}^m$ $(m \ll k)$

Background

- **Compressed sensing & Dictionary learning**

$$\min \ \|\alpha\|_0$$
$$s.t. \quad D\alpha = \mathbf{x}$$

sparse signal measurement

  - Measurement matrix (dictionary) $D \in \mathcal{R}^{m \times k}$
  - $\boldsymbol{\alpha} \in \mathcal{R}^k, \mathbf{x} \in \mathcal{R}^m$ $(m \ll k)$

- **Low-density parity check (LDPC) codes**

$$\min \ \|\mathbf{e}\|_0$$
$$s.t. \quad H\mathbf{e} = \mathbf{s}$$

errors in the signal parity check bits

  - Parity check matrix $H \in \{0,1\}^{m \times k}$
  - $\mathbf{e} \in \{0,1\}^k, \mathbf{s} \in \{0,1\}^m$ $(m \ll k)$

Background

- **Compressed sensing & Dictionary learning**

  $$\min \ \|\boldsymbol{\alpha}\|_0$$
  $$s.t. \quad D\boldsymbol{\alpha} = \mathbf{x}$$

  – Measurement matrix (dictionary) $D \in \mathcal{R}^{m\times k}$

  – $\boldsymbol{\alpha} \in \mathcal{R}^k, \mathbf{x} \in \mathcal{R}^m$ $(m \ll k)$

- **Low-density parity check (LDPC) codes**

  $$\min \ \|\mathbf{e}\|_0$$
  $$s.t. \quad H\mathbf{e} = \mathbf{s}$$

  – Parity check matrix $H \in \{0,1\}^{m\times k}$

  – $\mathbf{e} \in \{0,1\}^k, \mathbf{s} \in \{0,1\}^m$ $(m \ll k)$

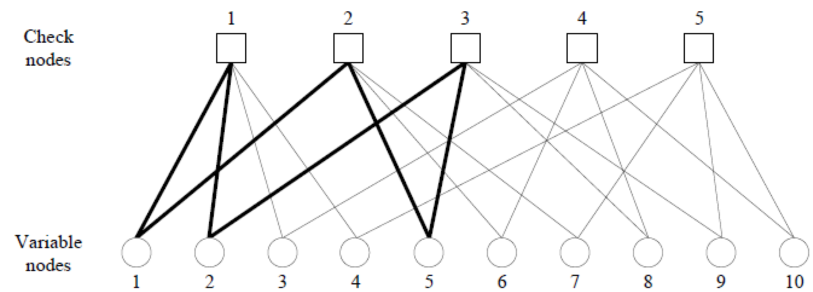
⟺ Bridge Lemma

Good parity check matrices are provably good for measurement matrices
(Dimakis *et al*. *IEEE T-IT*, 2012)

Background

LDPC codes & graphical models

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



- Key parameters
  1) Girth (shortest cycle)
  2) Degree distributions

Our Goal

Learn a good dictionary using LDPC codes

- **Step 1:** Initialize the dictionary with the LDPC code (sparse dictionary)
- **Step 2:** Update nonzero elements' values in the dictionary

➔ Less computational complexity

Step 1: Initialize the dictionary with the LDPC code

With the LDPC code, we fix the nonzero elements' locations in the dictionary

- *Progressive edge growth (PEG) algorithm:* Greedy algorithm to maximize girth
- *Density evolution (DE):* Find the optimized degree distributions of LDPC codes

- PEG: Hu *et al*. *IEEE T-IT*, 2005.
- DE: Richardson *et al*. IEEE T-IT, 2001.

Step 1: Initialize the dictionary with the LDPC code

With the LDPC code, we fix the nonzero elements' locations in the dictionary

- *Progressive edge growth (PEG) algorithm:* Greedy algorithm to maximize girth
- *Density evolution (DE):* Find the optimized degree distributions of LDPC codes

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{array}\right]$$

- PEG: Hu *et al*. *IEEE T-IT*, 2005.
- DE: Richardson *et al*. IEEE T-IT, 2001.

Step 2: Nonzero Elements' Values

- Initialization: Binary sparse matrix (by PEG and DE)
- Update nonzero elements' values (much less computations)

Algorithm 2 Dictionary Update. Mairal *et al.*, ICML 2009

Require: $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k] \in \mathbb{R}^{m \times k}$ (input dictionary),
$\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k] \in \mathbb{R}^{k \times k} = \sum_{i=1}^{t} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_i^T$,
$\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k] \in \mathbb{R}^{m \times k} = \sum_{i=1}^{t} \mathbf{x}_i \boldsymbol{\alpha}_i^T$.

1: **repeat**
2: **for** $j = 1$ to k **do**
3: Update the j-th column to optimize for (9):

$$\mathbf{u}_j \leftarrow \frac{1}{\mathbf{A}_{jj}}(\mathbf{b}_j - \mathbf{D}\mathbf{a}_j) + \mathbf{d}_j. \qquad (10)$$

$$\mathbf{d}_j \leftarrow \frac{1}{\max(||\mathbf{u}_j||_2, 1)} \mathbf{u}_j.$$

4: **end for**
5: **until convergence**
6: **Return D (updated dictionary).**

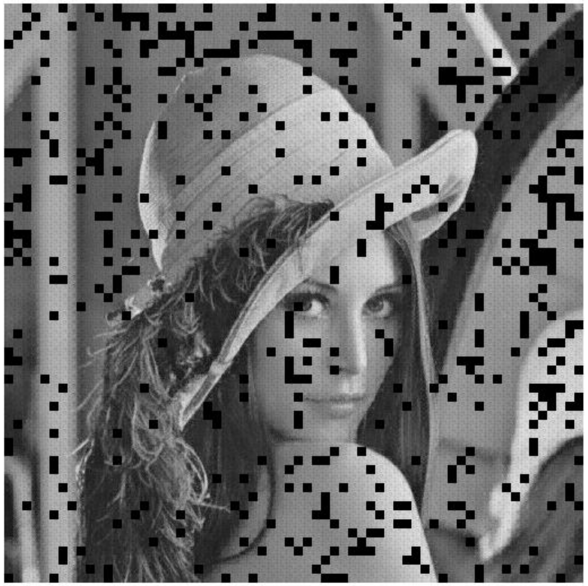
$$u_{i,j} \leftarrow \frac{1}{\mathbf{A}_{j,j}}\left(b_{i,j} - D_{i,:}a_{i,j}\right) + d_{i,j}$$

$u_{i,j} \neq 0$ for $i \in \mathcal{J}_j$ and
$u_{i',j} = 0$ for $i' \in \{1, \ldots, m\} \setminus \mathcal{J}_j$
$|\mathcal{J}_j| \ll m$

Simulation Results

- **Dictionary learning simulation with Lena image**
  → Learn the dictionary from the original image
  → Reconstruct the image from the learned dictionary

Original Image

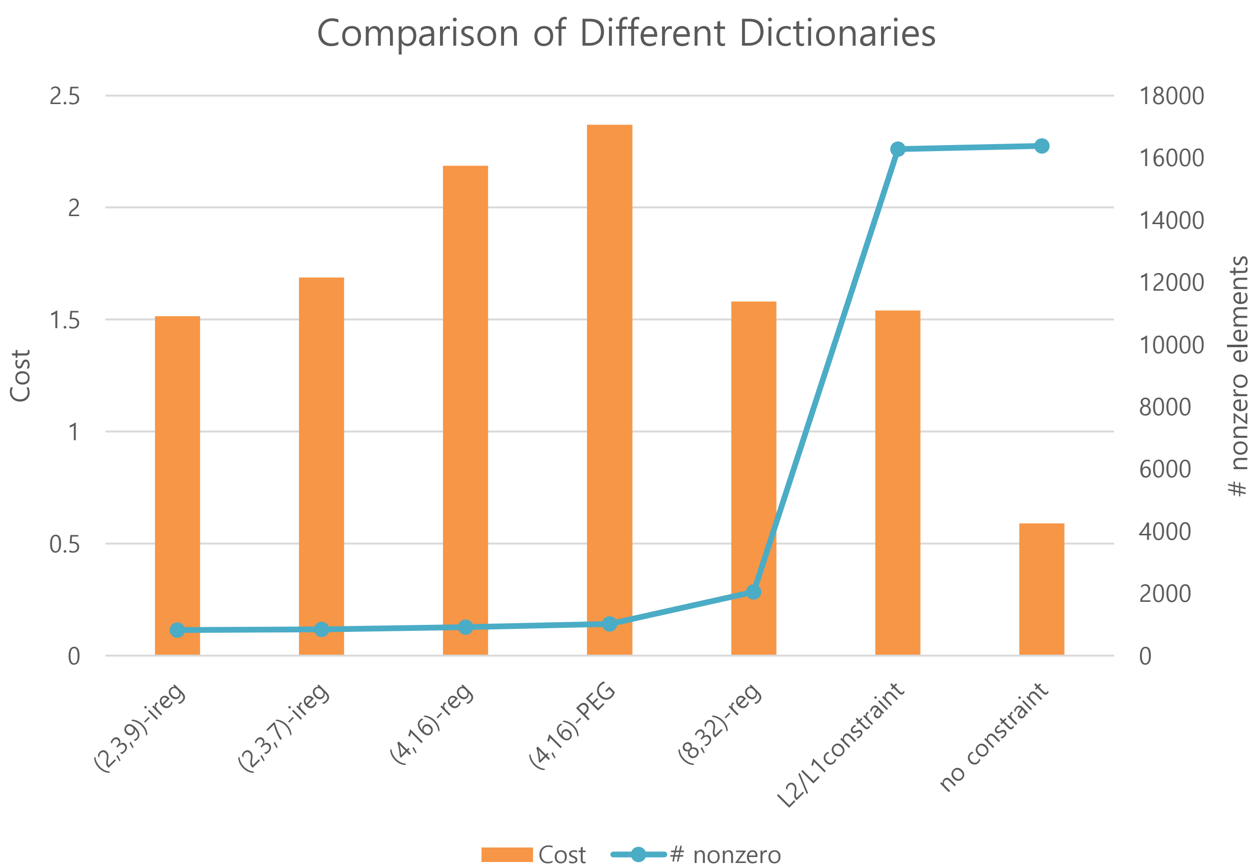


(2,3,9)-irregular PEG code

(2,3,9)-irregular PEG code after learning

Simulation Results

- **Comparison of Different Dictionaries**
  - → Cost = $0.5\sum||X - D\alpha||^2 + \lambda\sum|\alpha|$
  - → # of nonzero elements in the dictionary



Summary

- **Idea**
  - Construct a good sparse matrix for dictionary based on LDPC codes' design techniques
  - Update *only* dictionary's nonzero elements

- **Gain**
  - Less computational complexity (with comparable performance)
  - Less storage cost for the dictionary

Thanks.
Questions, or Suggestions?