Thresholding Begets Descent

Inderjit S. Dhillon
The University of Texas at Austin

Banff International Research Station
Jan 20, 2011

Joint work with Prateek Jain, Raghu Meka and Ambuj Tewari

Overview

- Hard Thresholding for Compressed Sensing
  - New Family of Algorithms with Guarantees
- Hard Thresholding for Matrix Completion
- Digression at End
  - Fast, Memory-Efficient Dimensionality Reduction of Massive Graphs

Compressed Sensing and Rank Minimization

$$(\mathbf{CS}): \quad \min_{\mathbf{x}} \quad \|\mathbf{x}\|_0 \quad \text{s.t} \quad A\mathbf{x} = b.$$

$$\mathbf{x} \in \mathbb{R}^n, \; A : \mathbb{R}^n \to \mathbb{R}^d, \; b \in \mathbb{R}^d.$$

$$(\mathbf{ARMP}): \quad \min_{X} \quad rank(X) \quad \text{s.t} \quad \mathcal{A}(X) = b.$$

$$X \in \mathbb{R}^{m \times n}, \; \mathcal{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^d, \; b \in \mathbb{R}^d.$$

- Can view CS as specific instance of ARMP with $X = Diag(x)$.

An Example: Minimum Rank Matrix Completion

- Netflix Challenge:
  - Given a few user-movie ratings
  - Goal: complete ratings matrix
- Small number of latent factors $\equiv$ low-rank
- Special case of ARMP:

$$
\begin{aligned}
(\mathbf{MCP}): \quad & \min_{X} \quad rank(X) \\
& \text{s.t} \quad \mathrm{tr}(X\mathbf{e}_j\mathbf{e}_i^T) = b_{ij}, \forall (i,j) \in \Omega.
\end{aligned}
$$

- Typically, number of samples very small: Netflix has 1% samples

CS and ARMP

Technique	CS	ARMP
Convex relaxation	ℓ_1 (Lasso)	Trace-norm (SVT)
Greedy approach	MP, OMP, CoSamp	ADMiRA
Hard Thresholding	IHT, GradeS	SVP, IHT

Table: CS vs ARMP

Restricted Isometry Property (RIP)

- Most CS methods assume RIP:

$$(1-\delta_k)\|\mathbf{x}\|^2 \le \|A\mathbf{x}\|^2 \le (1+\delta_k)\|\mathbf{x}\|^2,\ \forall \mathbf{x} \text{ s.t. } \|\mathbf{x}\|_0 \le k$$

- Generalization to ARMP:

$$(1-\delta_k)\|X\|_F^2 \le \|\mathcal{A}(X)\|_2^2 \le (1+\delta_k)\|X\|_F^2,\ \forall X \text{ s.t. } \operatorname{rank}(X) \le k$$

- Families satisfying RIP:

$$\mathcal{A}(X) = A\ vec(X),$$

  - $A_{ij} \sim \mathcal{N}(0, 1/d)$
  - $A_{ij} = \begin{cases} 1/\sqrt{d} & \text{with probability } 1/2 \\ -1/\sqrt{d} & \text{with probability } 1/2 \end{cases}$

Projected Gradient

- Consider

$$\min_{x} \psi(x) = \frac{1}{2}\|Ax - b\|_2^2,$$
$$\text{s.t } x \in \mathcal{C}(k) = \{x : supp(x) \leq k\}.$$

- Adapt classical projected gradient
- Efficient projection onto non-convex support constraint

Iterative Hard Thresholding (IHT)

Algorithm 1 IHT/GradeS Algorithm

Initialize $x^0 = 0,\ t = 0$
Set step size η_t
repeat
$$x^{t+1} = P_k(x^t - \eta_t \underbrace{A^*(Ax^t - b)}_{\nabla\psi(x)})$$
$t = t + 1$
until Convergence

IHT-Newton

Algorithm 2 IHT-Newton Algorithm

Initialize $x^0 = 0,\ t = 0$

Set step size η_t

repeat

$$y^{t+1} = P_k(x^t - \eta_t \underbrace{A^*(Ax^t - b)}_{\nabla\psi(x)})$$

$$x^{t+1} = \text{argmin}_{supp(x)=supp(y^{t+1})} \|Ax - b\|^2$$

$$t = t + 1$$

until Convergence

IHT/GradeS: Proof

- Simple analysis–apply RIP twice and Projection property once
- $\psi(x) = \frac{1}{2}\|Ax - b\|_2^2$

$$\psi(x^{t+1}) - \psi(x^t) = \langle \nabla\psi(x^t), x^{t+1} - x^t \rangle + \frac{1}{2}\|A(\overbrace{x^{t+1} - x^t}^{\text{supp } 2k})\|^2$$

IHT/GradeS: Proof

- Simple analysis–apply RIP twice and Projection property once
- $\psi(x) = \frac{1}{2}\|Ax - b\|_2^2$

$$\begin{aligned}\psi(x^{t+1}) - \psi(x^t) &= \langle \nabla\psi(x^t), x^{t+1} - x^t \rangle + \frac{1}{2}\|A(\overbrace{x^{t+1} - x^t}^{\text{supp } 2k})\|^2 \\ &\leq \langle \nabla\psi(x^t), x^{t+1} - x^t \rangle + \frac{1}{2}\underbrace{(1 + \delta_{2k})\|x^{t+1} - x^t\|^2}_{\text{Using RIP}},\end{aligned}$$

IHT/GradeS: Proof

- Simple analysis–apply RIP twice and Projection property once
- $\psi(x) = \frac{1}{2}\|Ax - b\|_2^2$

$$
\begin{aligned}
\psi(x^{t+1}) - \psi(x^t) &= \langle \nabla\psi(x^t), x^{t+1} - x^t \rangle + \frac{1}{2}\|A(\overbrace{x^{t+1} - x^t}^{\text{supp } 2k})\|^2 \\
&\leq \langle \nabla\psi(x^t), x^{t+1} - x^t \rangle + \frac{1}{2}\underbrace{(1+\delta_{2k})\|x^{t+1} - x^t\|^2}_{\text{Using RIP}}, \\
&= \frac{1}{2}(1+\delta_{2k})\|x^{t+1} - y^{t+1}\|^2 - \frac{1}{2(1+\delta_{2k})}\|A^T(Ax^t - b)\|^2
\end{aligned}
$$

IHT/GradeS: Proof

- Simple analysis–apply RIP twice and Projection property once
- $\psi(x) = \frac{1}{2}\|Ax - b\|_2^2$

$$\psi(x^{t+1}) - \psi(x^t) = \langle \nabla\psi(x^t), x^{t+1} - x^t \rangle + \frac{1}{2}\|A(\overbrace{x^{t+1} - x^t}^{\text{supp } 2k})\|^2$$

$$\leq \langle \nabla\psi(x^t), x^{t+1} - x^t \rangle + \frac{1}{2}\underbrace{(1+\delta_{2k})\|x^{t+1} - x^t\|^2}_{\text{Using RIP}},$$

$$= \frac{1}{2}(1+\delta_{2k})\|x^{t+1} - y^{t+1}\|^2 - \frac{1}{2(1+\delta_{2k})}\|A^T(Ax^t - b)\|^2$$

$$\text{where } y^{t+1} = x^t - \frac{1}{1+\delta_{2k}}\nabla\psi(x^t), \;\; x^{t+1} = P_k(y^{t+1})$$

IHT/GradeS: Proof

- Simple analysis–apply RIP twice and Projection property once
- $\psi(x) = \frac{1}{2}\|Ax - b\|_2^2$

$$
\begin{aligned}
\psi(x^{t+1}) - \psi(x^t) &= \langle \nabla\psi(x^t), x^{t+1} - x^t \rangle + \frac{1}{2}\|A(\overbrace{x^{t+1} - x^t}^{\text{supp } 2k})\|^2 \\
&\leq \langle \nabla\psi(x^t), x^{t+1} - x^t \rangle + \frac{1}{2}\underbrace{(1+\delta_{2k})\|x^{t+1} - x^t\|^2}_{\text{Using RIP}}, \\
&= \frac{1}{2}(1+\delta_{2k})\|x^{t+1} - y^{t+1}\|^2 - \frac{1}{2(1+\delta_{2k})}\|A^T(Ax^t - b)\|^2 \\
&\leq \frac{1}{2}(1+\delta_{2k})\underbrace{\|x^* - y^{t+1}\|^2}_{Projection} - \frac{1}{2(1+\delta_{2k})}\|A^T(Ax^t - b)\|^2
\end{aligned}
$$

IHT/GradeS: Proof

- Simple analysis–apply RIP twice and Projection property once
- $\psi(x) = \frac{1}{2}\|Ax - b\|_2^2$

$$\psi(x^{t+1}) - \psi(x^t) \leq \frac{1}{2}(1+\delta_{2k})\underbrace{\|x^* - y^{t+1}\|^2}_{Projection} - \frac{1}{2(1+\delta_{2k})}\|A^T(Ax^t - b)\|^2$$

$$= \langle \nabla\psi(x^t), x^* - x^t \rangle + \frac{1}{2}(1+\delta_{2k})\|x^* - x^t\|^2$$

IHT/GradeS: Proof

- Simple analysis–apply RIP twice and Projection property once
- $\psi(x) = \frac{1}{2}\|Ax - b\|_2^2$

$$\psi(x^{t+1}) - \psi(x^t) \leq \frac{1}{2}(1+\delta_{2k})\underbrace{\|x^* - y^{t+1}\|^2}_{Projection} - \frac{1}{2(1+\delta_{2k})}\|A^T(Ax^t - b)\|^2$$

$$= \langle \nabla\psi(x^t), x^* - x^t\rangle + \frac{1}{2}(1+\delta_{2k})\|x^* - x^t\|^2$$

$$\leq \langle \nabla\psi(x^t), x^* - x^t\rangle + \frac{1}{2}\underbrace{\frac{1+\delta_{2k}}{1-\delta_{2k}}\|A(x^* - x^t)\|^2}_{\text{Using RIP}}$$

IHT/GradeS: Proof

- Simple analysis–apply RIP twice and Projection property once
- $\psi(x) = \frac{1}{2}\|Ax - b\|_2^2$

$$\psi(x^{t+1}) - \psi(x^t) \leq \frac{1}{2}(1 + \delta_{2k}) \underbrace{\|x^* - y^{t+1}\|^2}_{Projection} - \frac{1}{2(1+\delta_{2k})}\|A^T(Ax^t - b)\|^2$$

$$= \langle \nabla\psi(x^t), x^* - x^t \rangle + \frac{1}{2}(1 + \delta_{2k})\|x^* - x^t\|^2$$

$$\leq \langle \nabla\psi(x^t), x^* - x^t \rangle + \frac{1}{2} \underbrace{\frac{1+\delta_{2k}}{1-\delta_{2k}}\|A(x^* - x^t)\|^2}_{\text{Using RIP}}$$

$$= \psi(x^*) - \psi(x^t) + \frac{\delta_{2k}}{(1-\delta_{2k})}\|A(x^* - x^t)\|^2,$$

IHT/GradeS: Proof

- Simple analysis–apply RIP twice and Projection property once
- $\psi(x) = \frac{1}{2}\|Ax - b\|_2^2$

$$\psi(x^{t+1}) - \psi(x^t) \le \frac{1}{2}(1+\delta_{2k})\underbrace{\|x^* - y^{t+1}\|^2}_{Projection} - \frac{1}{2(1+\delta_{2k})}\|A^T(Ax^t - b)\|^2$$

$$= \langle \nabla\psi(x^t), x^* - x^t\rangle + \frac{1}{2}(1+\delta_{2k})\|x^* - x^t\|^2$$

$$\le \langle \nabla\psi(x^t), x^* - x^t\rangle + \frac{1}{2}\underbrace{\frac{1+\delta_{2k}}{1-\delta_{2k}}\|A(x^* - x^t)\|^2}_{\text{Using RIP}}$$

$$= \psi(x^*) - \psi(x^t) + \frac{\delta_{2k}}{(1-\delta_{2k})}\|A(x^* - x^t)\|^2,$$

For exact case, $\psi(x^*) = 0,\ A(x^*) = b$. Hence,

$$\psi(x^{t+1}) \le \underbrace{\frac{2\delta_{2k}}{(1-\delta_{2k})}}_{<1 \text{ for } \delta_{2k}<1/3} \psi(x^t).$$

IHT: Recovery Guarantees

- Suppose $b = Ax^*$. When $\delta_{2k} < 1/3$, IHT outputs x st $\|Ax - b\|_2^2 \leq \epsilon$ in $\left\lceil C \log \frac{\|b\|^2}{2\epsilon} \right\rceil$ iterations [Garg & Khandekar, 2009].
- Similar geometric convergence for noisy case, $b = Ax^* + e$.
- Update: [Foucart, 2010] shows geometric convergence when $\delta_{3k} < 1/2$ with step size $\eta = 1$ (improved to $\delta_{3k} < 1/\sqrt{3}$ in [Foucart, 2011]).
- Similar guarantees for IHT-Newton, which empirically works better.

Structure of Gradient Step

- Consider IHT-Newton iterate x^t
- Let $supp(x^*) = S^*$, $supp(x^t) = S^t$, and $J = (S^* \cup S^t)^c$
- Since $A^*_{S^t}(Ax^t - b) = 0$, gradient step: $y_{t+1} = x^t - \eta_t A^*(Ax^t - b)$ has the form:

$$y_{t+1} = \begin{bmatrix} x_t \\ -\eta_t A^*_{S^* - S^t}(Ax^t - b) \\ -\eta_t A^*_J(Ax^t - b) \end{bmatrix}$$

New Algorithm — OMP(ℓ) with Replacement

Algorithm 3 OMPR(ℓ) Algorithm

Initialize $x^0 = A^* b$, $t = 0$
Set step size η_t
repeat
$$y^{t+1} = x^t - \eta_t \underbrace{A^*(Ax^t - b)}_{\nabla\psi(x)}$$
$$C = supp(x^t) \cup \text{Top } \ell \text{ indices of } y^{t+1}_{S_t^c}$$
$$y^{t+1} = P_k(y^{t+1}_C)$$
$$x^{t+1} = \text{argmin}_{supp(x)=supp(y^{t+1})} \|Ax - b\|^2$$
$$t = t + 1$$
until Convergence

- When $\ell = 1$, OMPR(1) replaces one "working-set" index at a time.
- Note: OMPR(k) is IHT-Newton.

Recovery Guarantees for OMPR(ℓ)

- Property 1: $\|y_D^{t+1}\| > \|x_U^t\|$, where "desired" index set $D = S^* - S_t$, and "undesired" index set $U = S_t - S^*$.
- Property 2: $\psi(x^t) - \psi(x^{t+1}) > c$
- Recovery Guarantee: Suppose $b = Ax^*$. When $\delta_{2k} < 1/2$ and $\eta = .9999$, OMPR(1) converges to x^* in $O(k)$ iterations.
- Similar guarantee for noisy case
- Gives "best-known" RIP guarantee
- Time complexity $= O(k * n * d)$, but least squares update at each iteration is cheap.

Matrix Completion

- Complete a low-rank matrix from few sampled entries
- Minimum rank matrix completion problem:

$$(\mathbf{MCP}): \min_{X} \ \text{rank}(X), \quad \text{s.t } \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(X^*).$$

- $\mathcal{P}_\Omega : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$—projection onto index set Ω, i.e.,

$$(\mathcal{P}_\Omega(X))_{ij} = \begin{cases} X_{ij} & \text{for } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

- Special case of ARMP: SVP can be applied directly
- Problem: MCP does not satisfy RIP in general

SVP: Hard Thresholding for Matrix Completion

$$\min_X \psi(X) = \frac{1}{2}\|P_\Omega(X - X^*)\|_F^2,$$
$$\text{s.t } X \in \mathcal{C}(k) = \{X : rank(X) \le k\}.$$

Algorithm 4 SVP for Matrix Completion

Initialize $X^0 = 0,\ t = 0$

Set step size $\eta_t = 1/(1+\delta)p$, p=sampling density, δ is a parameter

repeat

$X^{t+1} = P_k(X^t - \eta_t P_\Omega(X^t - X^*))$

$t = t + 1$

until Convergence

- $P_k(X) = U_k \Sigma_k V_k^T$: top k singular vectors of X
- Computation of k singular vectors of: $\underbrace{X^t}_{low\ \ rank} - \eta_t \underbrace{P_\Omega(X^t - X^*)}_{sparse}$
- Matrix-vector multiplication: $O((m+n)k + |\Omega|)$

SVP-Newton for Matrix Completion

Algorithm 5 SVP-Newton

Initialize $X^0 = 0,\ t = 0$
Set step size $\eta_t = 1/(1+\delta)p$, p=sampling density, δ is a parameter
repeat
  $Y^{t+1} = U_k \Sigma_k V_k^T$, where $svd(X^t - \eta_t P_\Omega(X^t - X^*)) = U \Sigma V^T$
  Given U_k and V_k, compute:

$$S_k = \underset{S}{\operatorname{argmin}} \|P_\Omega(U_k S V_k^T - X^*)\|_F^2$$

  $X^{t+1} = U_k S_k V_k^T$
  $t = t + 1$
until Convergence

OMPR(ℓ) for Matrix Completion

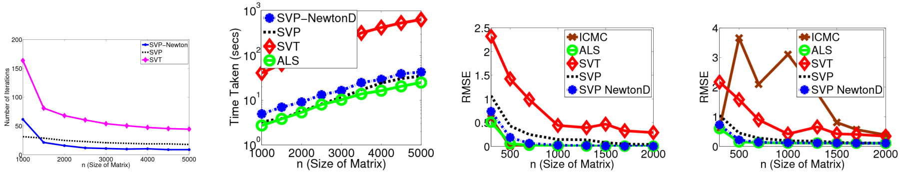
Algorithm 6 OMPR(ℓ) for Matrix Completion

Initialize $X^0 = U_k \Sigma_k V_k^T$, where $svd(P_\Omega(X^*)) = U \Sigma V^T$. Set $t = 0$
Set step size $\eta_t = 1/(1+\delta)p$, p=sampling density, δ is a parameter
repeat
Given $X_t = U_k \Sigma_k V_k^T$, compute:

$$S_k = \operatorname*{argmin}_S \|P_\Omega(U_k S V_k^T - X^*)\|_F^2$$

Compute svd of S_k to get svd of $U_k S_k V_k^T = \bar{U}_k \bar{\Sigma}_k \bar{V}_k^T$
Compute top ℓ singular triplets of $P_\Omega(\bar{U}_k \bar{\Sigma}_k \bar{V}_k^T - X^*)$ and take union with $\{\bar{U}_k, \bar{\Sigma}_k, \bar{V}_k^T\}$
Drop bottom ℓ singular triplets of the above set to obtain X^{t+1}
$t = t + 1$
until Convergence

Results: Matrix Completion for Synthetic Datasets



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

(a) Number of iterations required

(b) Time required for Noisy data

(c),(d) RMSE for Power-Law Sampling without/with Noise

Results: Matrix Completion for MovieLens Dataset

k	SVP-NewtonD	SVP	ALS	SVT
2	0.90	1.15	0.88	1.06
3	0.89	1.14	0.87	0.98
5	0.89	1.09	0.86	0.95
7	0.89	1.08	0.86	0.93
10	0.90	1.07	0.87	0.91
12	0.91	1.08	0.88	0.90

Table: RMSE obtained by various methods

- Problem: Ratings matrix is not sampled uniformly

Conclusions and Future Work

- New OMP with Replacement Algorithm
  - Guarantees recovery
  - Gives "best-known" RIP guarantee

- Future Work
  - Reduce Time Complexity (don't compute full gradient)
  - Explore IHT-Newton or OMPR(ℓ) in regression setting (when RIP does not hold)
  - Hard thresholding algorithms for other problems, e.g., sparse+low-rank matrix decomposition?

Paper will soon be available at: http://arxiv.org

Massive Social Networks

Testbed of Social Networks

Network	Date	# nodes	# links	# added links	% added links
Flickr	4/14/2007	1,990,149	41,302,536	–	–
	4/25/2007	1,990,149	42,056,754	754,218	1.8%
	5/6/2007	1,990,149	42,879,714	822,960	1.9%
LiveJournal	02/16/2009	1,770,961	83,663,478	–	–
	03/4/2009	1,770,961	84,413,542	750,064	0.8%
	04/03/2009	1,770,961	85,713,766	1,300,224	1.5%
MySpace	12/11/2008	2,137,264	90,333,122	–	–
	1/11/2009	2,137,264	90,979,264	646,142	0.7%
	2/14/2009	2,137,264	91,648,716	669,452	0.7%

	# clusters	avg size	% intra links	% inter links
Flickr	18	110,563	71.8%	28.2%
LiveJournal	17	106,241	72.5%	27.5%
MySpace	17	125,721	51.9%	48.1%

Motivation

- Need to compute matrix spectral functions, e.g. Katz measure for predicting future friendships

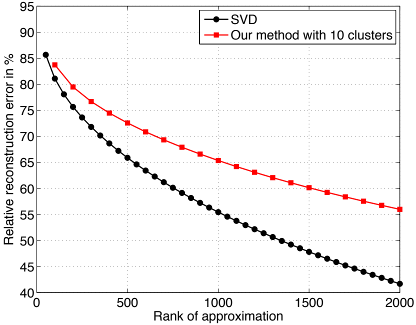
$$(I - \beta A)^{-1} = V(I - \beta \Lambda)^{-1} V^T$$

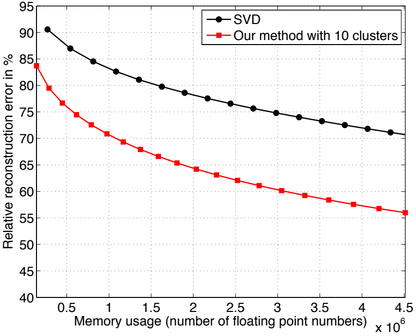
- Too expensive to compute for massive networks
- One solution: spectral approximation

$$f(A) \approx V f(\Lambda) V^T$$

- But SVD/PCA is wasteful (especially in terms of memory). Same for recent stochastic algorithms.

Empirical Results





Algorithm: Clustered low rank approximation

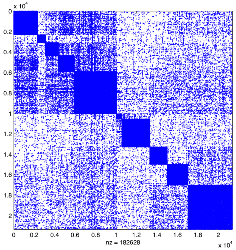
Require: An m**x**m adjacency matrix A of a graph, number of clusters c
Ensure: Clustered low rank approximation of A

1: Cluster the graph into c clusters
2: Compute a low rank approximation of each cluster

$$U_i S_i V_i \approx A_{ii}$$

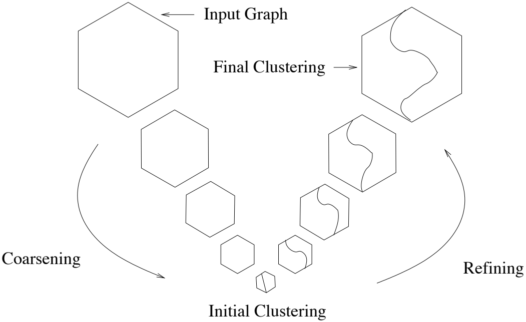
3: Extend the cluster-wise approximations, into an approximation for the entire matrix A

$$S_{ij} = U_i^{\mathsf{T}} A_{ij} V_j$$



Graph Clustering: Multilevel Approach

- Overview:



[CHACO, Hendrickson & Leland, 1994]

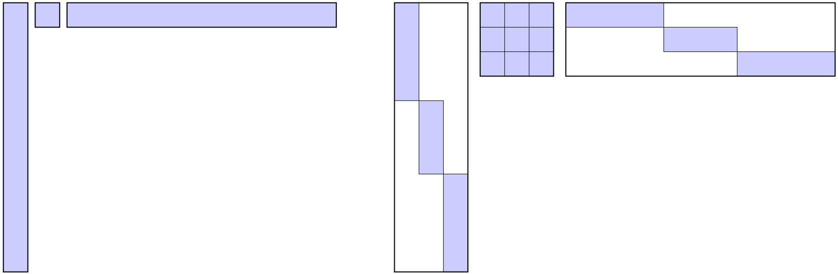
[METIS, Karypis & Kumar, 1999]

[GRACLUS, Dhillon, Guan & Kulis, 2005]

Low rank vs clustered low-rank

Low rank: $A \approx U\Sigma V^T$

Clustered low rank: $A \approx \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}^T$



- Observe $\text{diag}(U_1, U_2, \cdots, U_c)$ and has the same memory usage as U