KDD2018

FASTEN: Fast Sylvester Equation Solver for Graph Mining

Presenter: Boxin Du

Joint work by:

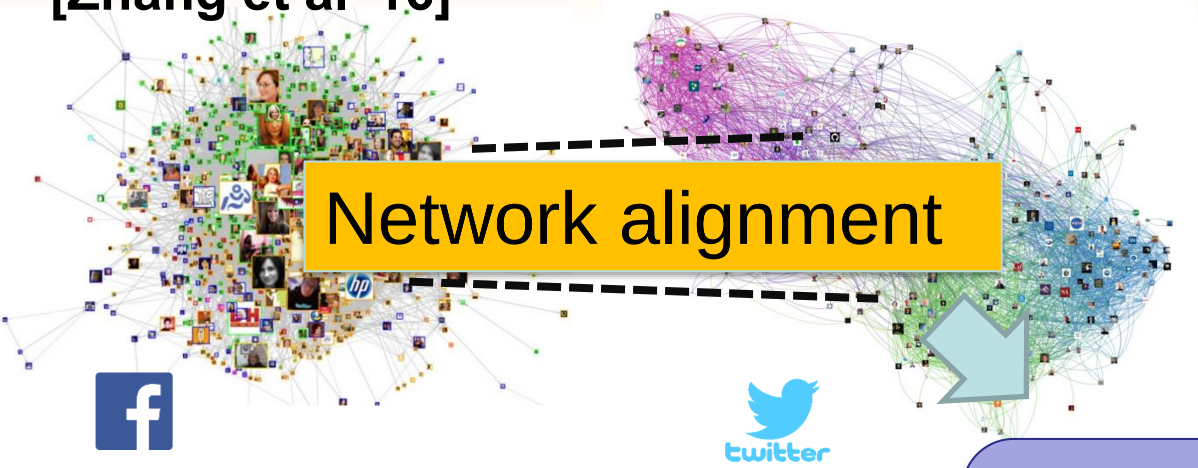
Boxin Du

Hanghang Tong

Why Sylvester Equation?

- Link users from different social networks

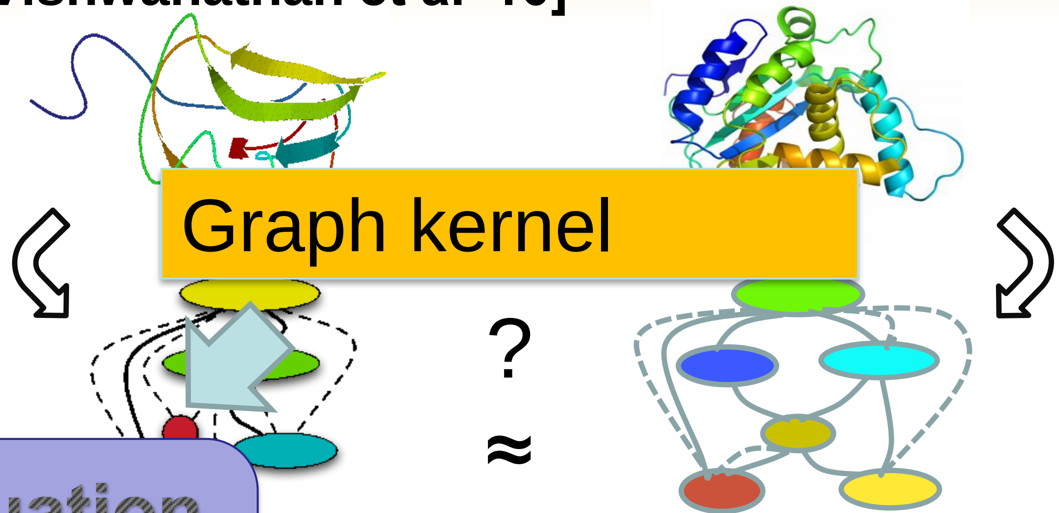
[Zhang et al’ 16]



Network alignment

- Protein function prediction

[Vishwanathan et al’ 10]



Graph kernel

Sylvester Equation

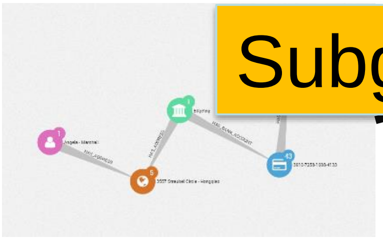
- Fraudulent transaction pattern ma

[Du et al’ 17]

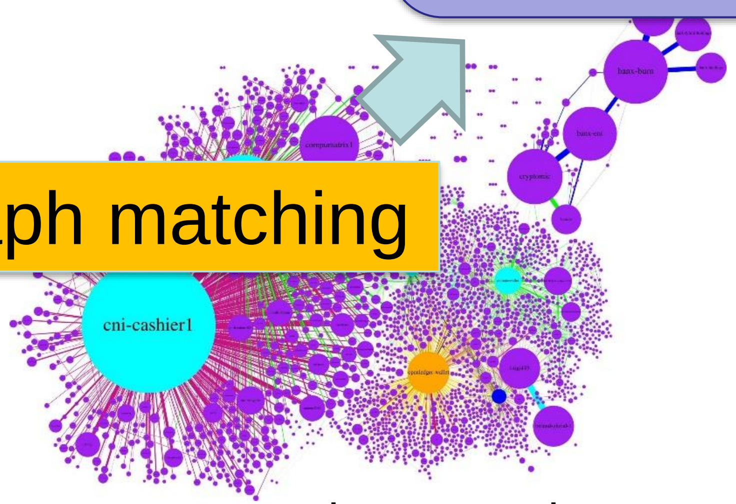
l similarity calculation

[Yu-Chen al’ 15]

Subgraph matching

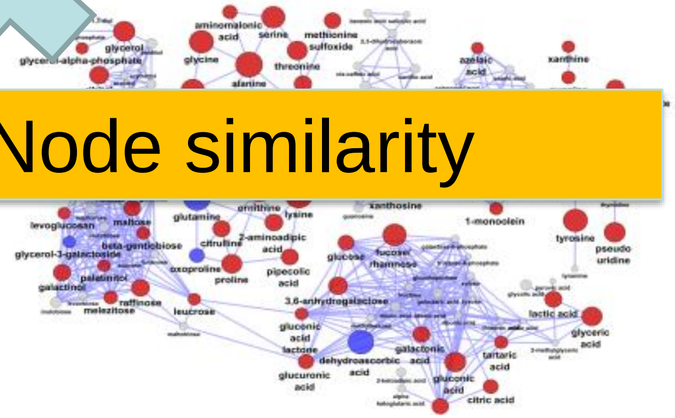


Query pattern



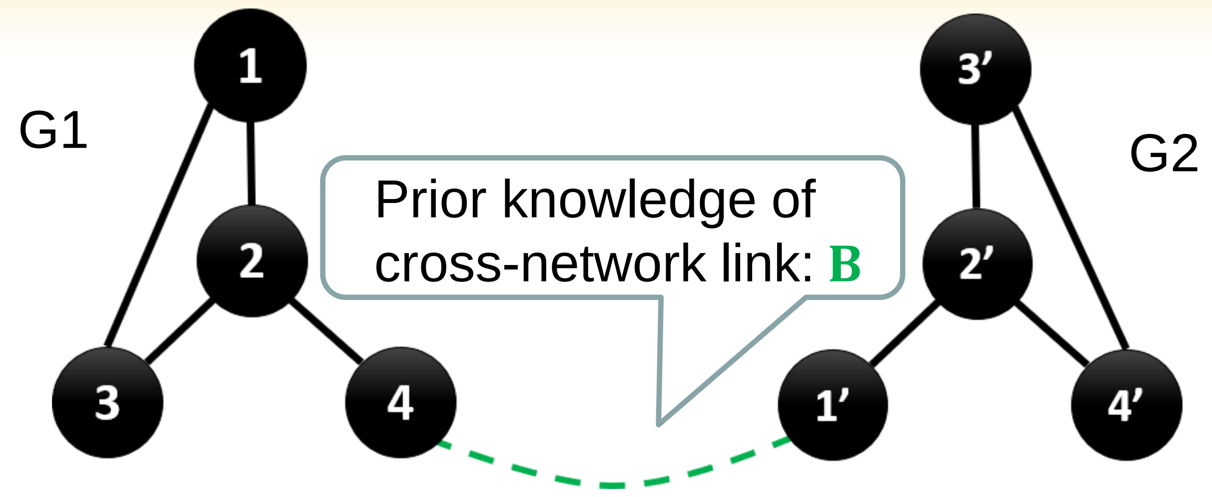
Transaction network

Node similarity



Chemical compound network

What is Sylvester Equation: an example on plain graph



Adjacency matrix $\mathbf{A}_1$:

0	1	1	0
1	0	1	1
1	1	0	0
0	1	0	0

$\mathbf{B}$:

0	0	0	0
0	0	0	0
0	0	0	0
1	0	0	0

Adjacency matrix: $\mathbf{A}_2$:

0	1	0	0
1	0	1	1
0	1	0	1
0	1	1	0

	1'	2'	3'	4'
1	0.040	0.047	0.048	0.048
2	0.027	0.138	0.047	0.047
3	0.040	0.047	0.048	0.048
4	0.261	0.027	0.040	0.040

Solution matrix $\mathbf{X}$ of the Sylvester equation:

$$\mathbf{X} = \mathbf{A}_2\mathbf{X}\mathbf{A}_1 + \mathbf{B}$$

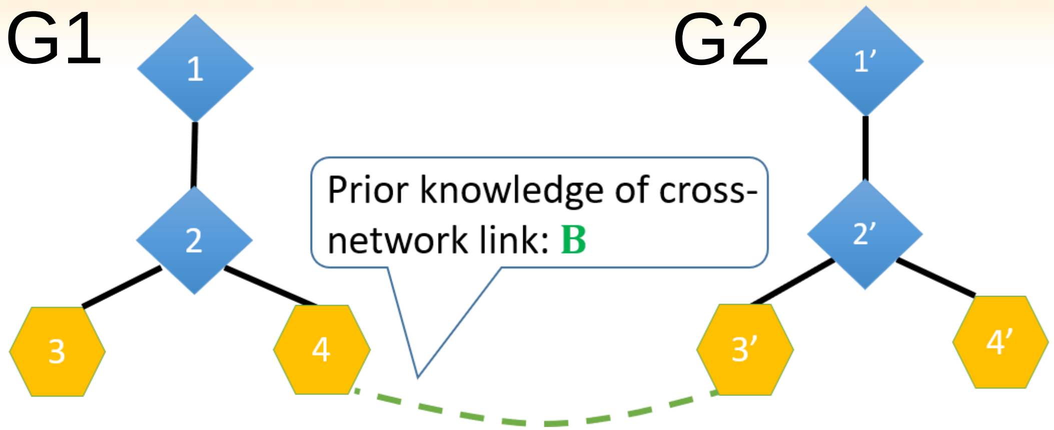
($\mathbf{A}_1$ and $\mathbf{A}_2$ are normalized)

- Sylvester equation $\mathbf{X} = \mathbf{A}_2\mathbf{X}\mathbf{A}_1 + \mathbf{B}$ gives the cross-network node similarity matrix $\mathbf{X}$;

[1] Zhang, Si, and Hanghang Tong. "Final: Fast attributed network alignment." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

[2] Singh, Rohit, Jinbo Xu, and Bonnie Berger. "Global alignment of multiple protein interaction networks with application to functional orthology detection." *Proceedings of the National Academy of Sciences* (2008).

What is Sylvester Equation: an example on attributed graph



Input graphs with node attributes (colors and shapes)

	1'	2'	3'	4'
1	0.022	0	0	0
2	0	0.083	0	0
3	0	0	0.022	0.022
4	0	0	0.222	0.022

Solution $\mathbf{X}$ of $\mathbf{X} - \sum_{i,j=1}^{2} \mathbf{A}_2^{(ij)} \mathbf{X} (\mathbf{A}_1^{(ij)})^{\mathrm{T}} = \mathbf{B}$

($\mathbf{A}_1$ and $\mathbf{A}_2$ are normalized)

e.g. $\mathbf{A}_1^{11}$:

0		0	0
	0	0	0
0	0	0	0
0	0	0	0

$\mathbf{A}_1^{12}$:

0	0	0	0
0	0		
0	0	0	0
0	0	0	0

$\mathbf{A}_1$:

0	1	0	0
1	0	1	1
0	1	0	0
0	1	0	0

$\mathbf{B}$:

0	0	0	0
0	0	0	0
0	0	0	0
0	0	1	0

$\mathbf{A}_2$:

0	1	0	0
1	0	1	1
0	1	0	0
0	1	0	0

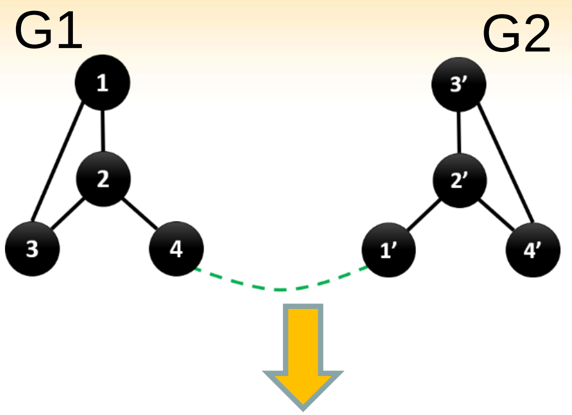
- Sylvester equation $\mathbf{X} - \sum_{i,j=1}^{2} \mathbf{A}_2^{(ij)} \mathbf{X} (\mathbf{A}_1^{(ij)})^{\mathrm{T}} = \mathbf{B}$ gives the cross-network node similarity matrix $\mathbf{X}$;

[1] Zhang, Si, and Hanghang Tong. "Final: Fast attributed network alignment." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
[2] Singh, Rohit, Jinbo Xu, and Bonnie Berger. "Global alignment of multiple protein interaction networks with application to functional orthology detection." *Proceedings of the National Academy of Sciences* (2008).

Formal Definition of Sylvester Equation (Plain Graph)

- **Given:**
  - Two graphs G_1 and G_2 (the adjacency matrices are $\mathbf{A_1}$ and $\mathbf{A_2}$);
  - The preference matrix $\mathbf{B}$.
- **Find:** the solution $\mathbf{X}$ of Sylvester equation: $\mathbf{X} - \mathbf{A_2} \mathbf{X} \mathbf{A_1}^T = \mathbf{B}$

  or $\mathbf{x}$ of its equivalent linear system: $(\mathbf{I} - \mathbf{W})\mathbf{x} = \mathbf{b}$
- **Mathematical details:**
  - $\mathbf{A_1} \leftarrow \alpha^{1/2} \mathbf{D_1}^{-1/2} \mathbf{A_1} \mathbf{D_1}^{-1/2}$, $\mathbf{A_2} \leftarrow \alpha^{1/2} \mathbf{D_2}^{-1/2} \mathbf{A_2} \mathbf{D_2}^{-1/2}$;
  - $\mathbf{D_1}$ and $\mathbf{D_2}$ are the diagonal degree matrices of $\mathbf{A_1}$ and $\mathbf{A_2}$, $0 < \alpha < 1$;
  - $\mathbf{W} = \mathbf{A_1} \otimes \mathbf{A_2}$ (both are normalized), $\mathbf{x} = \text{vec}(\mathbf{X})$, $\mathbf{b} = \text{vec}(\mathbf{B})$.



	1'	2'	3'	4'
1	0.040	0.047	0.048	0.048
2	0.027	0.138	0.047	0.047
3	0.040	0.047	0.048	0.048
4	0.261	0.027	0.040	0.040

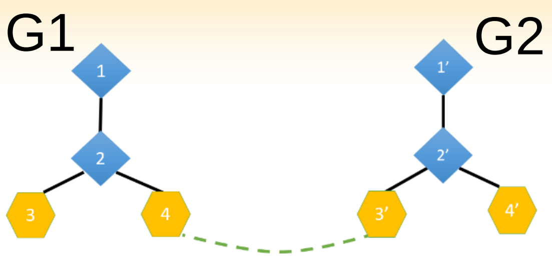
Solution matrix $\mathbf{X}$

[1] Zhang, Si, and Hanghang Tong. "Final: Fast attributed network alignment." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
[2] Singh, Rohit, Jinbo Xu, and Bonnie Berger. "Global alignment of multiple protein interaction networks with application to functional orthology detection." *Proceedings of the National Academy of Sciences* (2008).

Formal Definition of Sylvester Equation (Attributed Graph)

- **Given:**
  - Two graphs $G_1 = \{A_1, N_1\}$, $G_2 = \{A_2, N_2\}$;
  - The preference matrix $\mathbf{B}$.

$N_2^j(a,a) = 1$ if node a has node attribute j, o/w it is zero.



- **Find:** the solution $\mathbf{X}$ of Sylvester equation: $\mathbf{X} - \sum_{i,j=1}^{l} \mathbf{A}_2^{(ij)} \mathbf{X} (\mathbf{A}_1^{(ij)})^{\mathbf{T}} = \mathbf{B}$

  or $\mathbf{x}$ of its equivalent linear system: $\left[\mathbf{I} - \sum_{i,j=1}^{l} (\mathbf{A}_1^{(ij)} \otimes \mathbf{A}_2^{(ij)})\right] \mathbf{x} = \mathbf{b}$

	1'	2'	3'	4'
1	0.022	0	0	0
2	0	0.083	0	0
3	0	0	0.022	0.022
4	0	0	0.222	0.022

Solution matrix $\mathbf{X}$

- **Mathematical details:**
  - $\mathbf{A}_1^{(ij)} \leftarrow \alpha^{1/2} \mathbf{D}_1^{-1/2} \mathbf{N}_1^i \mathbf{A}_1 \mathbf{N}_1^j \mathbf{D}_1^{-1/2}$, $\mathbf{A}_2^{(ij)} \leftarrow \alpha^{1/2} \mathbf{D}_2^{-1/2} \mathbf{N}_2^i \mathbf{A}_2 \mathbf{N}_2^j \mathbf{D}_2^{-1/2}$;
  - $\mathbf{A}_1^{(ij)}$ is the adjacency matrix 'filtered' by attribute *i* and *j*.
  - l : the number of node attributes, $\mathbf{x}$ = vec($\mathbf{X}$), $\mathbf{b}$ = vec($\mathbf{B}$).

$\mathbf{A}_1^{11}$:

0		0	0
	0	0	0
0	0	0	0
0	0	0	0

[1] Zhang, Si, and Hanghang Tong. "Final: Fast attributed network alignment." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
[2] Singh, Rohit, Jinbo Xu, and Bonnie Berger. "Global alignment of multiple protein interaction networks with application to functional orthology detection." *Proceedings of the National Academy of Sciences* (2008).

Challenges of Solving the Sylvester Equation

- **Size of $\mathbf{A_1} \otimes \mathbf{A_2}$:**
  - $n^2 \times n^2$ (for plain graphs with n nodes and m edges);
  - Straightforward solver costs $O(n^6)$ (time) and $O(m^2)$ (space);
  - State-of-the-art methods: time complexity at least $O(mn + n^2)$;

The $\Omega(n^2)$ bottleneck

- **With node attributes:**
  - Add additional $O(l)$ complexity (for l discrete node attributes);
- **Size of solution matrix $\mathbf{X}$:**
  - $n \times n$;
  - Usually not sparse;
  - Limit the time/space complexity of the equation solver.

The $\Omega(n^2)$ bottleneck

Comparison of Methods for the Sylvester Equation

Algorithm	Attributed (Y/N)	Exact Solution (Y/N)	Time Complexity	Space Complexity	
Fixed Point (FP) [Vishwanathan et al' 10]	✓	✓	$O(n^3)$	$O(m^2)$	Traditional methods
Conjugate Gradient (CG) [Y Saad et al' 03]	✓	✓	$O(n^3)$	$O(m^2)$	Traditional methods
Sylv. [Vishwanathan et al' 10]	✓	✓	$O(n^3)$	$O(m^2)$	Traditional methods
ARK [U Kang et al' 12]	✓	✗	$O(n^2)$	$O(n^2)$	Recent methods
Cheetah [L Li et al' 10]	✓	✗	$O(rn^2)$	$O(n^2)$	Recent methods
NI-Sim [C Li et al' 10]	✗	✗	$O(n^2)$	$O(r^2n^2)$	Recent methods
FINAL-P [S Zhang et al'16]	✗	✓	$O(mn + n^2)$	$O(n^2)$	Recent methods
FINAL-NE [S Zhang et al'16]	✓	✓	$O(lmn + ln^2)$	$O(n^2)$	Recent methods
FINAL-N+ [S Zhang et al'16]	✓	✗	$O(n^2)$	$O(n^2)$	Recent methods
FASTEN-P	✗	✓	$O(kn^2)$	$O(n^2)$	This Paper
FASTEN-P+	✗	✓	$O(km + k^2n)$	$O(m + kn)$	This Paper
FASTEN-N	✓	✓	$O(mn/l + kn^2/l)$	$O(m/l + n^2)$	This Paper
FASTEN-N+	✓	✓	$O(km + k^2ln)$	$O(m + kln)$	This Paper

- **Obs.: Tradition methods:** all attributed and provide exact solution, but high complexity; **Recent methods:** at least $O(n^2)$, and are often approximated/not attributed.
- **Q:** Can we have a solution that is **attributed**, **exact**, and **more efficient?**

Roadmap

- Motivations ✓
- Background
- Proposed Algorithms for plain graphs
- Proposed Algorithms for attributed graphs
- Experimental Results
- Conclusions

Krylov Subspace Method (KSM) for Linear System

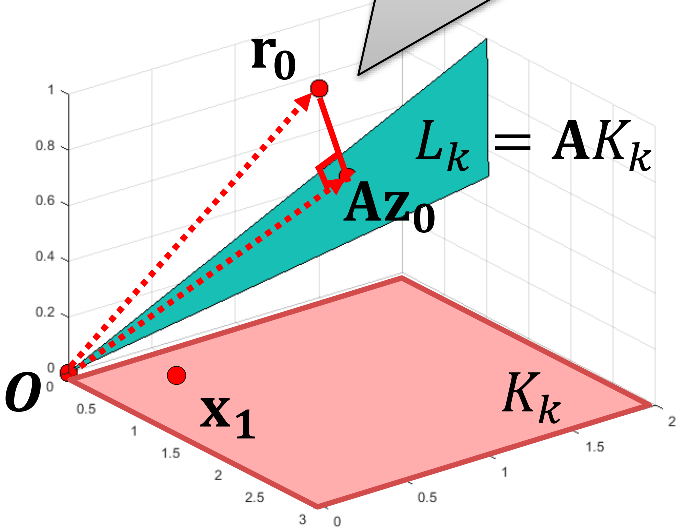
- Minimal residual method for linear system $\mathbf{Ax} = \mathbf{b}$ ($\mathbf{x} \in \mathrm{R}^n$):
  - Extract $\mathbf{x}$ from k-dimensional subspace of R^n ⟹ $\mathbf{x} \in \mathbf{x_0} + K_k$
  - Minimize residual $\mathbf{r} = \mathbf{b} - \mathbf{Ax} \perp L_k$ ⟹ small scaled system
  - Iteratively update $\mathbf{x}$ and $\mathbf{r}$ until $|\mathbf{r}|_2$ is small enough

$\mathbf{r_0} - \mathbf{Az_0}$: new residual
L_k: Subspace of constraints

- Example:

$$\underbrace{\begin{bmatrix} 3 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 3 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{\mathbf{b}} \Longrightarrow \underbrace{\begin{bmatrix} 6 & 6 \\ 11 & 6 \end{bmatrix}}_{\mathbf{A'}} \underbrace{\begin{bmatrix} x_1' \\ x_2' \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 4 \\ 5 \end{bmatrix}}_{\mathbf{c}}$$

  - Let $\mathbf{x_0} = [\mathbf{0}, \mathbf{0}, \mathbf{0}]^T$ (i.e. $\mathbf{r_0} = \mathbf{b}$), extract $\mathbf{x_1} \in K_k$.
  - Let $\mathbf{x_1} = \mathbf{x_0} + \mathbf{z_0}$, minimize $\mathbf{r} = \mathbf{r_0} - \mathbf{Az_0}$ ($\mathbf{r} \perp L_k$).
  - Update $\mathbf{x}$ in 3-d space.

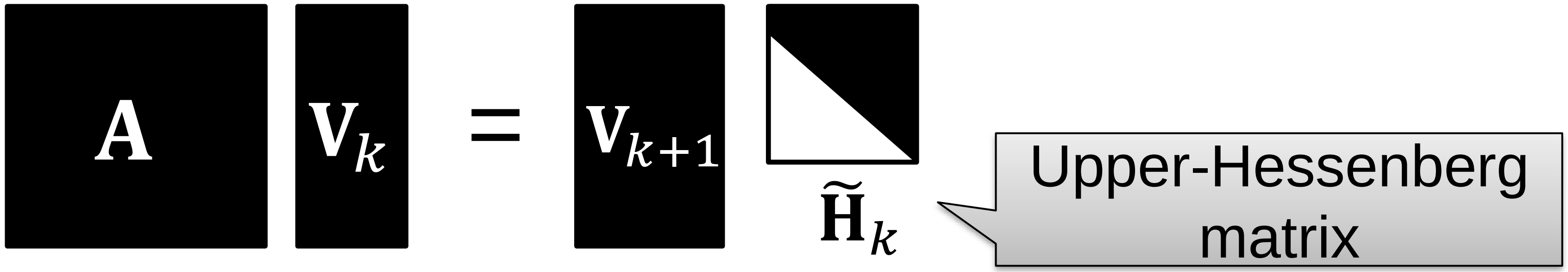


[1] Saad, Yousef. *Iterative methods for sparse linear systems*. Vol. 82. siam, 2003.

KSM for Linear System (cont'd)

- Krylov subspace:
  - $K_k(\mathbf{A}, \mathbf{r_0}) = span\{\mathbf{r_0}, \mathbf{Ar_0}, \mathbf{A}^2\mathbf{r_0}, \dots, \mathbf{A}^{k-1}\mathbf{r_0}\}$;

  - Arnoldi process outputs i orthonormal basis: $\mathbf{V}_i = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_i], i \in \{k, k+1\}$
  - $\mathbf{AV}_k = \mathbf{V}_{k+1}\widetilde{\mathbf{H}}_k$

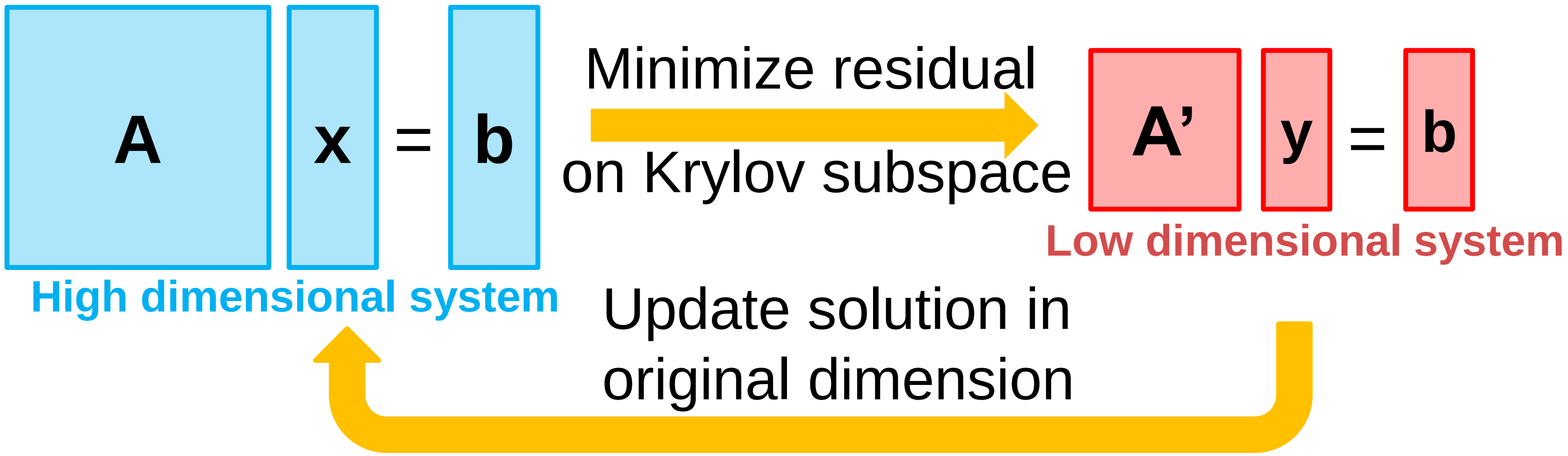


- Krylov subspace-based Minimal Residual method:
  - Extract solution from k-dimensional Krylov subspace (let $K_k = K_k(\mathbf{A}, \mathbf{r_0})$);
  - Minimize the residual $\mathbf{r}$ and update solution at every iteration.

[1] Saad, Yousef. *Iterative methods for sparse linear systems*. Vol. 82. siam, 2003.

Advantages of KSM with Minimal Residual

- Arnoldi: $O(m)$ for sparse system;
- Solve small scaled system every iteration;
- Exact solution, no approximation needed;
- Upper-Hessenberg makes solving system faster.

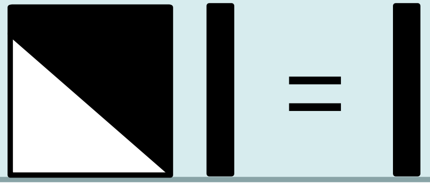


- Details:
  Minimize residual

$$J(\mathbf{y}) = ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2$$
$$= ||\mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{V}_k\mathbf{y})||_2$$
$$= \cdots$$
$$= ||\beta\mathbf{e}_1 - \widetilde{\mathbf{H}}_k\mathbf{y}||_2$$

equivalent to solve:

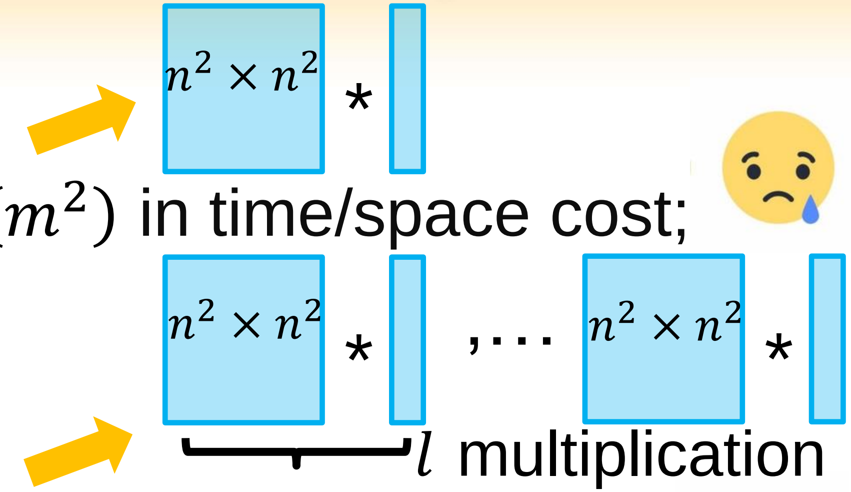
$$\widetilde{\mathbf{H}}_k\mathbf{y} = \beta\mathbf{e}_1$$



[1] Saad, Yousef. *Iterative methods for sparse linear systems*. Vol. 82. siam, 2003.

Challenges of Applying KSM on Sylvester Equation

- **Size of $(\mathbf{I}-\mathbf{W})\mathbf{x}=\mathbf{b}$:**
  – Generate $K_{k^2}(\mathbf{I} - \mathbf{A}_1 \otimes \mathbf{A}_2,\ \mathbf{r}_0)$ ⟶ $O(n^4)$ or $O(m^2)$ in time/space cost;

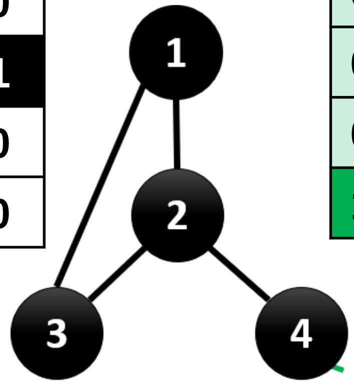


- **Size of $\left[\mathbf{I} - \sum_{i,j=1}^{l}(\mathbf{A}_1^{(ij)} \otimes \mathbf{A}_2^{(ij)})\right]\mathbf{x} = \mathbf{b}$:**
  – Generate $K_{k^2}(\mathbf{I} - \sum_{i,j=1}^{l}(\mathbf{A}_1^{(ij)} \otimes \mathbf{A}_2^{(ij)}), \mathbf{r}_0)$ ⟶ $O(ln^4)$ or $O(lm^2)$ cost.

- **Example:**

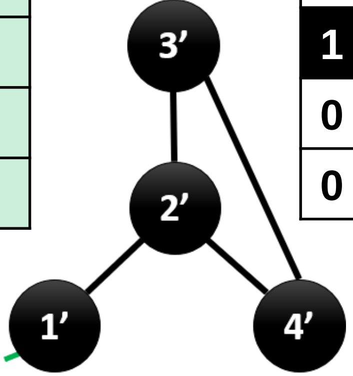
0	1	1	0
1	0	1	1
1	1	0	0
0	1	0	0

$\mathbf{A}_1$



0	0	0	0
0	0	0	0
0	0	0	0
1	0	0	0

$\mathbf{B}$



0	1	0	0
1	0	1	1
0	1	0	1
0	1	1	0

$\mathbf{A}_2$

⟶ Krylov subspace of $K_{k^2}(\mathbf{I} - \mathbf{A}_1 \otimes \mathbf{A}_2, \mathbf{r}_0)$: 16 dimension

Roadmap

- Motivations ✓
- Background ✓
- Proposed Algorithms for plain graphs
- Proposed Algorithms for attributed graphs
- Experimental Results
- Conclusions

Key Ideas

- #1: Kronecker Krylov Subspace (KKS)
  - Implicit construction of the original large Krylov subspace
  - Largely reduce the time/space complexity $O(n^4) \Rightarrow O(n^2)$
- #2: MRES* on KKS with Implicit Solution Representation
  - Solve small scaled system and update solution till converge
  - Further reduce the time/space complexity $O(kn^2) \Rightarrow O(k^2n + km)$

*: MRES: Minimal Residual method

Kronecker Krylov Subspace (Details)

$(\mathbf{I}-\mathbf{W})\mathbf{x}=\mathbf{b}$

- **Step 1:** Choose Arnoldi vectors $\mathbf{g}$, $\mathbf{f}$; $O(n^2)$
- **Step 2:** Generate $K_k(\mathbf{A_1},\mathbf{g})$; $O(km)$
- **Step 3:** Generate $K_k(\mathbf{A_2},\mathbf{f})$; $O(km)$
- Details:
  - Choosing $\mathbf{g}$, $\mathbf{f}$ s.t. $\mathbf{r_0} \in K_k(\mathbf{A_1},\mathbf{g}) \otimes K_k(\mathbf{A_2},\mathbf{f})$:
  - If $||\mathbf{R_0}||_1 \leq ||\mathbf{R_0}||_\infty$,

    $\mathbf{f}$: $\mathbf{R_0}$'s column of largest norm, $\mathbf{g}=\mathbf{R_0^T}\mathbf{f}/|\mathbf{f}|_2^2$
  - If $||\mathbf{R_0}||_1 > ||\mathbf{R_0}||_\infty$,

    $\mathbf{g}$: $\mathbf{R_0}$'s row of largest norm, $\mathbf{f}=\mathbf{R_0^T}\mathbf{g}/|\mathbf{g}|_2^2$

$K_k(\mathbf{A_1},\mathbf{g}) \otimes K_k(\mathbf{A_2},\mathbf{f})$

$\mathbf{A_1}\mathbf{V}_k=\mathbf{V}_{k+1}\widetilde{\mathbf{H}}_1$

$\mathbf{V}_k=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_k]$

$\mathbf{A_2}\mathbf{W}_k=\mathbf{W}_{k+1}\widetilde{\mathbf{H}}_2$

$\mathbf{W}_\mathrm{k}=[\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_k]$

Theorem: $\mathbf{V}_k \otimes \mathbf{W}_\mathrm{k}$ forms the orthonormal basis of the Kronecker Krylov subspace; don't need to be computed directly

Example

$(\mathbf{I}-\mathbf{W})\mathbf{x}=\mathbf{b}$

- **Step 1:** $\mathbf{R_0}=\mathbf{B}\ (\mathbf{x_0}=\mathbf{0})$, $\mathbf{f}=[0,0,0,1]^T$, $\mathbf{g}=[1,0,0,0]^T$;
- **Step 2:** $K_k(\mathbf{A_1},\mathbf{g})=span\{[0,0.7071,0.7071,0],^T\ [1,0,0,0]^T\}$;

  $\mathbf{v_1}$ $\mathbf{v_2}$

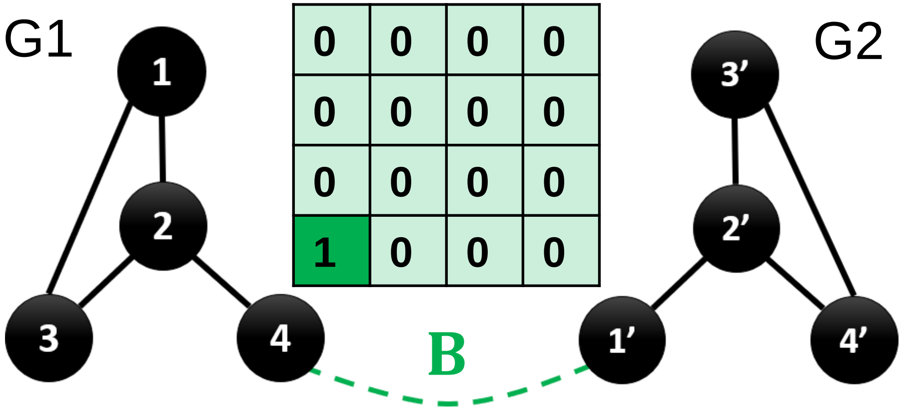
- **Step 3:** $K_k(\mathbf{A_2},\mathbf{f})=span\{[0,0.7071,0.7071,0]^T,[0,0,0,1]^T\}$;

  $\mathbf{w_1}$ $\mathbf{w_2}$

- $K_k(\mathbf{A_1},\mathbf{g})\otimes K_k(\mathbf{A_2},\mathbf{f})=span\{\mathbf{v_1}\otimes\mathbf{w_1},\mathbf{v_1}\otimes\mathbf{w_2},\mathbf{v_2}\otimes\mathbf{w_1},\mathbf{v_2}\otimes\mathbf{w_2}\}$.

0	1	1	0
1	0	1	1
1	1	0	0
0	1	0	0

A1



0	0	0	0
0	0	0	0
0	0	0	0
1	0	0	0

0	1	0	0
1	0	1	1
0	1	0	1
0	1	1	0

A2

Minimal Residual (Details):

$(\mathbf{I}-\mathbf{W})\mathbf{x} = \mathbf{b}$

- **Step 1:** Initial residual: $\mathbf{r_0} = \mathbf{b} - (\mathbf{I} - \alpha\mathbf{W})\mathbf{x_0}$
- **Step 2:** Let new solution:

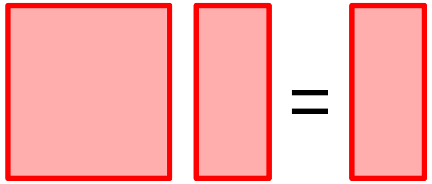
$$\mathbf{x} = \mathbf{x_0} + \mathbf{z_0}, \mathbf{z_0} \in K_k(\mathbf{A_1}, \mathbf{g}) \otimes K_k(\mathbf{A_2}, \mathbf{f})$$

- **Step 3:** Minimize new residual:

$$||\mathbf{R}||_2 = \left|\left| \mathbf{W}_{k+1}^{\mathbf{T}}\mathbf{R_0}\mathbf{V}_{k+1} - \widetilde{\mathbf{H}}_2\mathbf{Y}\widetilde{\mathbf{H}}_1^{\mathbf{T}} + \mathbf{I}_{k+1,k}\mathbf{Y}\mathbf{I}_{k+1,k}^{T} \right|\right|_{\mathbf{F}}$$

$-\mathbf{C}$ $\quad$ $\mathcal{L}(\mathbf{Y})$

Small system $\mathcal{L}(\mathbf{Y}) = \mathbf{C}$



- Both $\mathbf{Y}$ and $\mathbf{C}$ are k by k: small scaled system.
- **Step 4:** Update solution $\mathbf{X}$ and residual $\mathbf{R}$.

$$\mathbf{X} \leftarrow \mathbf{X} + \mathbf{V_k}\mathbf{Y}\mathbf{W_k^T}, \mathbf{R} \leftarrow \mathbf{R} - \mathbf{V_{k+1}}\widetilde{\mathbf{H}}_1\mathbf{Y}\widetilde{\mathbf{H}}_2^{\mathbf{T}}\mathbf{W_{k+1}^T} + \mathbf{V_k}\mathbf{Y}\mathbf{W_k^T}$$

Effectiveness: this method gives the exact solution of the Sylvester equation on plain graphs w.r.t. a tolerance ϵ.

Complexity: Time: $O(kn^2)$, Space: $O(n^2)$

Easy to solve!, $k \ll n$

FASTEN-P

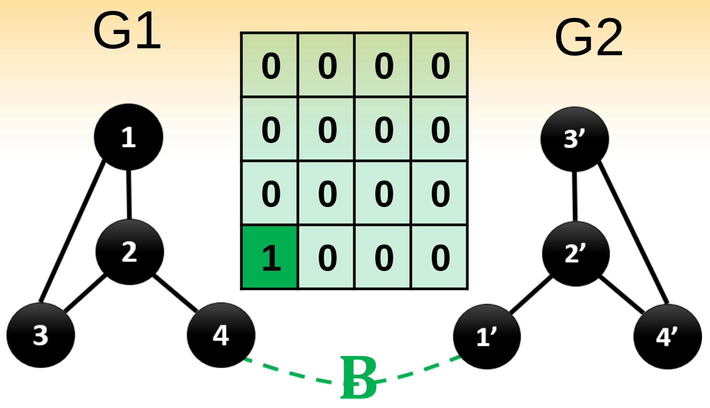
- Major steps:

1. $O(n^2)$ Choosing Arnoldi vectors $\mathbf{g}$ and $\mathbf{f}$ by $\mathbf{R}$
2. → $O(km)$ Arnoldi Process on $\mathbf{A}_1$ and $\mathbf{A}_2$ to get the orthogonal basis $\mathbf{V}_k$, $\mathbf{W}_k$
3. → $O(iter * k^3)$ Solve the new linear system in low dimensional space $L(\mathbf{Y}) = \mathbf{C}$
4. → $O(kn^2)$ Update $\mathbf{X}$ and $\mathbf{R}$ by $\mathbf{Y}$ and check stopping condition

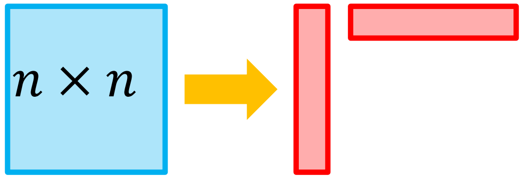
Till converge ($||\mathbf{R}||_F < \epsilon$, e.g. 10^{-8}).

- Details:
  - $\mathbf{X}$ is often initialized as $\mathbf{0}$, and $\mathbf{R} = \mathbf{B}$;
  - **Overall Complexity:** time: $O(kn^2)$; space: $O(n^2)$;

Can we further scale up? $\mathbf{X} - \mathbf{A}_2\mathbf{X}\mathbf{A}_1{}^T = \mathbf{B}$



- Goal: complexity: from $O(n^2)$ to linear
- Difficulties:
  - $\mathbf{X}$: $n \times n$, $O(n^2)$ seems to be the lower bound;
  - $\mathbf{X}$: in general not sparse.
- Observation:
  - $\mathbf{B}$ is often sparse and low-rank (sparse anchor links across network);
  - If prior anchor links are unknown: $\mathbf{B}$ is uniform (rank 1);
  - $\mathbf{B}$ is low-rank ➡ $\mathbf{X}$ must have low-rank property (see proof in paper).
- Solution:
  - Implicit representation of residual $\mathbf{R}$, intermediate solution $\mathbf{X}$.



Kronecker Krylov Subspace with Low-rank Residual

- **Step 1:** Represent $\mathbf{R_0}$ by low-rank matrices $\mathbf{U_1}, \mathbf{U_2}$: $O(n)$.
- **Step 2:** Choose Arnoldi vectors $\mathbf{g}$, $\mathbf{f}$: $O(rn)$ (r: rank of $\mathbf{U_1}, \mathbf{U_2}$)
- **Step 3:** Generate $K_k(\mathbf{A_1}, \mathbf{g})$, $K_k(\mathbf{A_2}, \mathbf{f})$: $O(km)$, and obtain

$$\mathbf{A_1 V}_k = \mathbf{V}_{k+1}\widetilde{\mathbf{H}}_1$$
$$\mathbf{V}_k = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$$

$$\mathbf{A_2 W}_k = \mathbf{W}_{k+1}\widetilde{\mathbf{H}}_2$$
$$\mathbf{W}_\mathrm{k} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k]$$

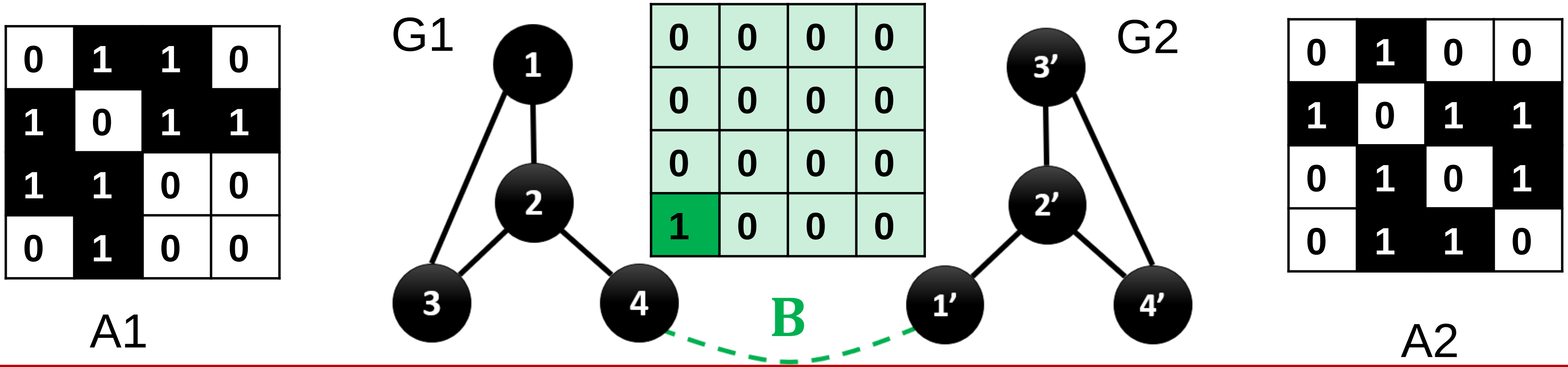
- Details:
  - Choosing $\mathbf{g}$, $\mathbf{f}$ (let $\mathbf{r_1} = \mathbf{e}^\mathrm{T}\mathbf{U_1 U_2}, \mathbf{r_2} = \mathbf{U_1 U_2 e}$):
  - If $\max(\mathbf{r_1}) \geq \max(\mathbf{r_2})$,

  $\mathbf{f} = \mathbf{U_1 U}_2(:, i_1)$, $\mathbf{g} = \mathbf{U}_2^\mathrm{T}\mathbf{U}_1^\mathrm{T}\mathbf{f}/|\mathbf{f}|_2^2$ ($i_1$is the index of $\mathbf{r_1}$'s largest entry)
  - If $\max(\mathbf{r_1}) < \max(\mathbf{r_2})$,

  $\mathbf{g} = \mathbf{U}_2^T\mathbf{U_1}(i_2, :)$, $\mathbf{f} = \mathbf{U_1 U_2 g}/|\mathbf{g}|_2^2$ ($i_2$is the index of $\mathbf{r_2}$'s largest entry)

Example

- **Step 1:**
  - **B**: Assume each node in G1 has at most one 1-to-1 anchor link to G2.
  - $\mathbf{R_0} = \mathbf{B} = [0,0,0,1]^{\mathbf{T}} * [1,0,0,0] = \mathbf{U_1 U_2}$; $O(n)$.
- **Step 2:** Choose Arnoldi vectors, $\mathbf{f} = [0,0,0,1]^T$, $\mathbf{g} = [1,0,0,0]^T$; $O(rn)$
- **Step 3:**
  - $K_k(\mathbf{A_1}, \mathbf{g}) = span\{[0,0.7071,0.7071,0],^T\ [1,0,0,0]^T\}$; $O(km)$
  - $K_k(\mathbf{A_2}, \mathbf{f}) = span\{[0,0.7071,0.7071,0]^T, [0,0,0,1]^T\}$; $O(km)$



Minimal Residual Method with Low-rank Representation

- **Step 1:** Obtain and solve small scaled system $\mathcal{L}(\mathbf{Y}) = \mathbf{C}$.
- **Step 2:** Implicit solution representation $\mathbf{P} = [\mathbf{P}, \mathbf{V_k}\mathbf{Y}], \mathbf{Q} = [\mathbf{Q}, \mathbf{W_k^T}]$;

(Original updating: $\mathbf{X} \leftarrow \mathbf{X} + \mathbf{V_k}\mathbf{Y}\mathbf{W_k^T}$)

Low-rank property: If $\mathbf{B}$ is rank r, the rank of $\mathbf{X}$ is upper-bounded by $iter * r$ ($iter$: the iteration number)

- **Step 3:** Let $L_2 = V_{k+1}\widetilde{\mathbf{H}}_1\mathbf{Y}\widetilde{\mathbf{H}}_2^T$, $\mathbf{P_2} = \mathbf{W_{k+1}^T}$, $\mathbf{L_3} = \mathbf{V_k}\mathbf{Y}$, $\mathbf{P_3} = \mathbf{W_k^T}$

Construct new residual $\mathbf{U_1} = [\mathbf{U_1}, \mathbf{L_2}, \mathbf{L_3}], \mathbf{U_2} = \left[\mathbf{U_2^T}, \mathbf{P_2^T}, \mathbf{P_3^T}\right]^{\mathbf{T}}$

(Original updating: $\mathbf{R} \leftarrow \mathbf{R} - \mathbf{V_{k+1}}\widetilde{\mathbf{H}}_1\mathbf{Y}\widetilde{\mathbf{H}}_2^{\mathbf{T}}\mathbf{W_{k+1}^T} + \mathbf{V_k}\mathbf{Y}\mathbf{W_k^T}$)

Complexity:
Time: $O(k(k+2)n)$,
Space: $O(m+kn)$

FASTEN-P+

- Major steps:

1. $O(rn)$ Choosing Arnoldi vectors $\mathbf{g}$ and $\mathbf{f}$ by $\mathbf{U_1}$, $\mathbf{U_2}$
2. $O(km)$ Arnoldi Process on $\mathbf{A_1}$ and $\mathbf{A_2}$ to get the orthogonal basis $\mathbf{V}_k$, $\mathbf{W}_k$
3. $O(iter * k^3)$ Solve the new linear system in low dimensional space $L(\mathbf{Y}) = \mathbf{C}$
4. $O(k^2 n)$ Update $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{U_1}$, $\mathbf{U_2}$ by $\mathbf{Y}$ and check stopping condition

Till converge ($||\mathbf{R}||_F < \epsilon$)

- Details:
  - 4 : $||\mathbf{R}||_F$ can be computed as $trace(\mathbf{U}_2^T(\mathbf{U}_1^T\mathbf{U}_1)\mathbf{U}_2)$;
  - **Overall Complexity:** time: $O(km + k^2 n)$; space: $O(m + kn)$;

Roadmap

- Motivations ✓
- Background ✓
- Proposed Algorithms for plain graphs ✓
- Proposed Algorithms for attributed graphs
- Experimental Results
- Conclusions

Key Ideas

- #1: Decomposition of Sylvester equation
  - Decompose the equation to a inter-correlated Sylvester equation set
  - Each decomposed equation is small-scaled & fast to solve
- #2: Apply FASTEN-P(+) on decomposed equation
  - Apply Block Coordinate Descent (BCD) on the whole equation set
  - Efficiently solve every single equation by FASTEN-P(+)

Decomposition of Sylvester Equation

	1′	2′	3′	4′
1	0.022	0	0	0
2	0	0.083	0	0
3	0	0	0.022	0.022
4	0	0	0.222	0.022

Solution matrix $\mathbf{X}$

- Observation: $\mathbf{X} - \sum_{i,j=1}^{l} \mathbf{A}_2^{(ij)} \mathbf{X} (\mathbf{A}_1^{(ij)})^{\mathbf{T}} = \mathbf{B}$
  - The solution matrix $\mathbf{X}$ has block-diagonal structure
  - The equation can be decomposed to:

$$\begin{cases} \mathbf{X}^{ii} - \sum_{\mathbf{q}=1}^{\mathbf{l}} \mathbf{A}_2^{iq} \mathbf{X}^{qq} \left(\mathbf{A}_1^{iq}\right)^{\mathbf{T}} = \mathbf{B}^{ii} \\ \mathbf{X}^{ij} = \mathbf{B}^{ij} \qquad (1 \leq i,j \leq l, i \neq j) \end{cases}$$

Diagonal block variables

Off-diagonal block variables

  - $\mathbf{A}_1^{iq}$ is a block of $\mathbf{A}_1$ of rows from attribute i to columns of attribute q.
  - **Off-diagonal block: need not to be solved**
  - **Diagonal block: apply Block Coordinate Descent (BCD)**

Apply FASTEN-P(+) on Decomposed Equation

- Observation:

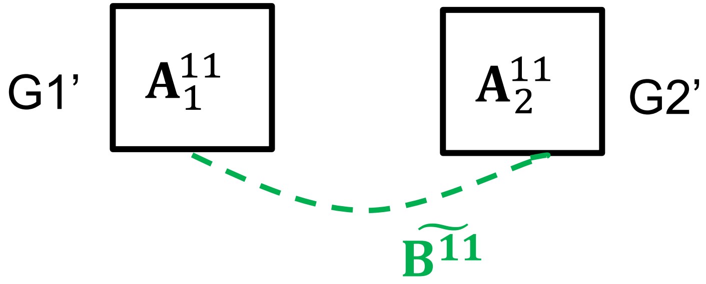
$$\mathbf{X}^{ii} - \sum_{\mathbf{q=1}}^{\mathbf{l}} \mathbf{A}_{\mathbf{2}}^{iq}\mathbf{X}^{qq}\left(\mathbf{A}_{\mathbf{1}}^{iq}\right)^{\mathbf{T}} = \mathbf{B}^{ii}$$

Diagonal block variables

  - When applying BCD: solve a non-attributed Sylvester equation each time
  - e.g.: when solving $\mathbf{X}^{11}$, the equation becomes:

$$\mathbf{X}^{11} - \mathbf{A}_2^{11}\mathbf{X}^{11}(\mathbf{A}_1^{11})^T = \mathbf{B}^{11} + \sum_{q\neq 1}^{l} \mathbf{A}_2^{1q}\mathbf{X}^{qq}(\mathbf{A}_1^{1q})^T = \widetilde{\mathbf{B}^{11}}$$

  - Apply FASTEN-P(+) to solve the above equation.



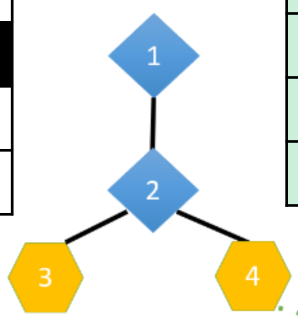
Example

- In this example, the attributed Sylvester equation is decomposed to:

0	1	0	0
1	0	1	1
0	1	0	0
0	1	0	0

A1

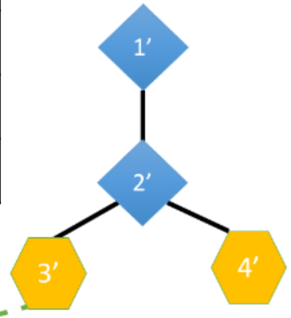
G1



0	0	0	0
0	0	0	0
0	0	0	0
0	0	1	0

$\mathbf{B}$

G2



0	1	0	0
1	0	1	1
0	1	0	0
0	1	0	0

A2

	1'	2'	3'	4'
1	0.022	0	0	0
2	0	0.083	0	0
3	0	0	0.022	0.022
4	0	0	0.222	0.022

Solution matrix $\mathbf{X}$

$\mathbf{X}^{11}$ and $\mathbf{X}^{22}$ are two 2 by 2 diagonal blocks

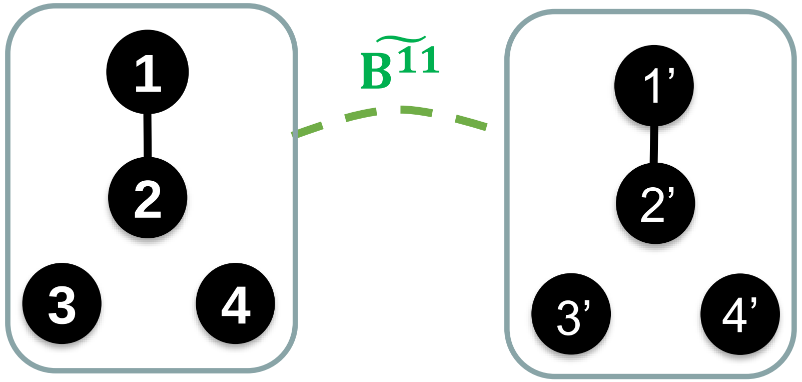
$$\mathbf{X} - \sum_{i,j=1}^{2} \mathbf{A}_2^{(ij)} \mathbf{X} (\mathbf{A}_1^{(ij)})^{\mathrm{T}} = \mathbf{B}$$

$$\begin{cases} \mathbf{X}^{11} - [\mathbf{A}_2^{11}\mathbf{X}^{11}(\mathbf{A}_1^{11})^T + \mathbf{A}_2^{12}\mathbf{X}^{22}(\mathbf{A}_1^{12})^T] = \mathbf{B}^{11} \\ \mathbf{X}^{22} - [\mathbf{A}_2^{21}\mathbf{X}^{11}(\mathbf{A}_1^{21})^T + \mathbf{A}_2^{22}\mathbf{X}^{22}(\mathbf{A}_1^{22})^T] = \mathbf{B}^{22} \\ \mathbf{X}^{12} = \mathbf{B}^{12} \\ \mathbf{X}^{21} = \mathbf{B}^{21} \end{cases}$$

Diagonal

Off-diagonal

e.g. Solve non-attributed Sylvester equation on $\mathbf{A}_1^{11}$, $\mathbf{A}_2^{11}$ for $\mathbf{X}^{11}$:



DATA Lab

Arizona State University

FASTEN-N

$$\mathbf{X} - \sum_{i,j=1}^{l} \mathbf{A}_2^{(ij)} \mathbf{X} (\mathbf{A}_1^{(ij)})^{\mathrm{T}} = \mathbf{B}$$

- Major steps:

1. Initialize each diagonal $\mathbf{X}^{ii}$ and the residual $\mathbf{R}$
2. $O(m)$ Construct block matrices $\mathbf{A}_1^{ij}$, $\mathbf{A}_2^{ij}$, $\mathbf{B}^{ij}$ by $\mathbf{N}_1$, $\mathbf{N}_2$
3. $O(mn/l)$ Iterate l times to solve l block variables by BCD & FASTEN-P
4. $O(kn^2/l)$ Update $\mathbf{X}$ and $\mathbf{R}$; check stopping condition

Till converge ($||\mathbf{R}||_F < \epsilon$)

- Details:
  - (2) : $\mathbf{N}_1$, $\mathbf{N}_2$ are the node attribute matrices of $\mathbf{A}_1$ and $\mathbf{A}_2$.
  - **Overall Complexity**: time: $O(mn/l + kn^2/l)$; space: $O(m/l + n^2)$;

From FASTEN-N to FASTEN-N+

$$\mathbf{X} - \sum_{i,j=1}^{l} \mathbf{A}_2^{(ij)} \mathbf{X} (\mathbf{A}_1^{(ij)})^{\mathbf{T}} = \mathbf{B}$$

- Major steps:

1. $O(n)$ Initialize each implicit solution $\mathbf{P}_i, \mathbf{Q}_i$ and the residual $\mathbf{U_1}, \mathbf{U_2}$
2. $O(m)$ Construct block matrices $\mathbf{A}_1^{ij}$, $\mathbf{A}_2^{ij}$, $\mathbf{B}^{ij}$ by $\mathbf{N_1}$, $\mathbf{N_2}$
3. $O(km)$ Iterate l times to solve l block variables by BCD & FASTEN-P+
4. $O(k^2 ln)$ Update $\mathbf{P}_i, \mathbf{Q}_i$ and $\mathbf{U_1}, \mathbf{U_2}$; check stopping condition

Till converge ($||\mathbf{R}||_F < \epsilon$)

- Details:
  - Key idea: apply FASTEN-P+ instead of FASTEN-P in step 3;
  - **Overall Complexity**: time: $O(km + k^2 ln)$; space: $O(m + kln)$;

DATA Lab

Roadmap

- Motivations ✓
- Background ✓
- Proposed Algorithms for plain graphs ✓
- Proposed Algorithms for attributed graphs ✓
- Experimental Results
- Conclusions

Experimental Setup

- Datasets Summary:

Dataset Name	Category	# of Nodes	# of Edges
DBLP	Co-authorship	9,143	16,338
Flickr	User relationship	12,974	16,149
LastFm	User relationship	15,436	32,638
Aminer	Academic network	1,274,360	4,756,194
LinkedIn	Social network	6,726,290	19,360,690

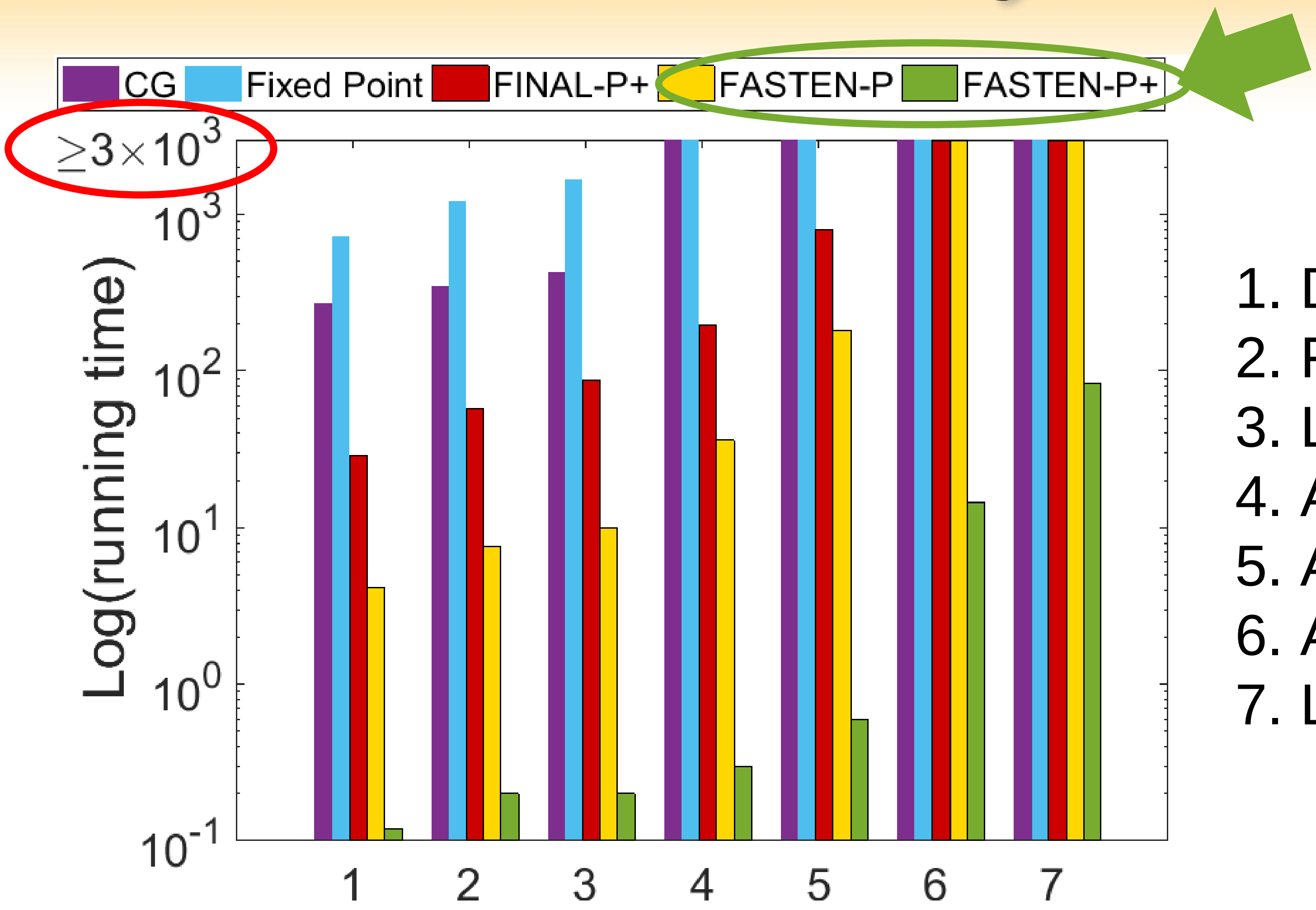
- Baseline methods
  - *Conjugate Gradient method (CG)* [Saad Y. SIAM 03]
  - *Fixed Point (FP)* [Saad Y. SIAM 03]

  } Exact methods

  - *FINAL-P+ & FINAL-N+* [Zhang et al. KDD'16]

  } Approximated methods

Experimental Result - Efficiency

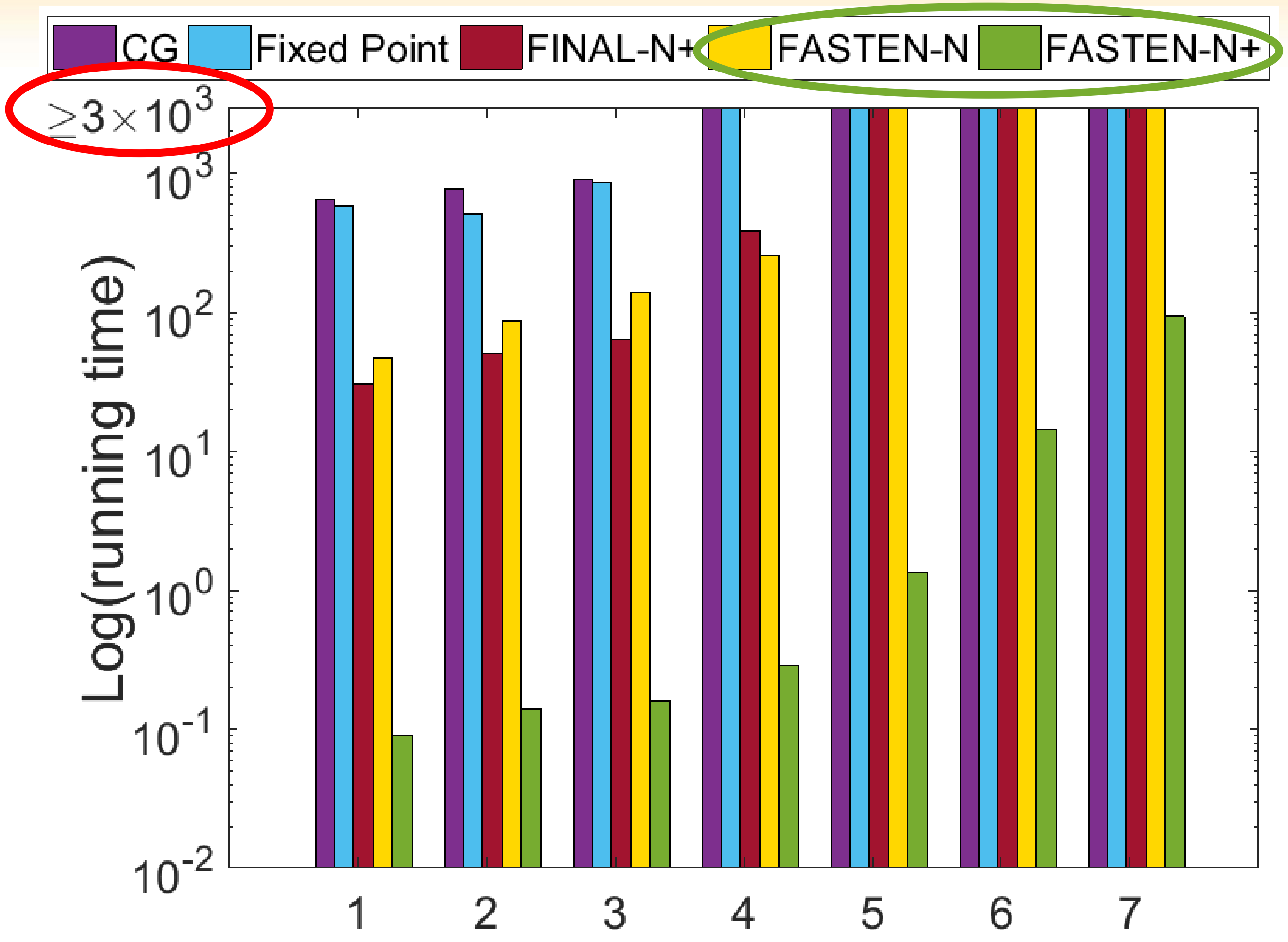


1. DBLP (9,143 nodes)
2. Flickr (12.974 nodes)
3. LastFm (15,436 nodes)
4. Aminer with 25K nodes
5. Aminer with 100K nodes
6. Aminer with 1.2M nodes
7. LinkedIn (6.7M nodes)

- Obs.: maximum speedup: > 10,000 times with 25K-node network.
- Better than approximated methods!

Experimental Result - Efficiency

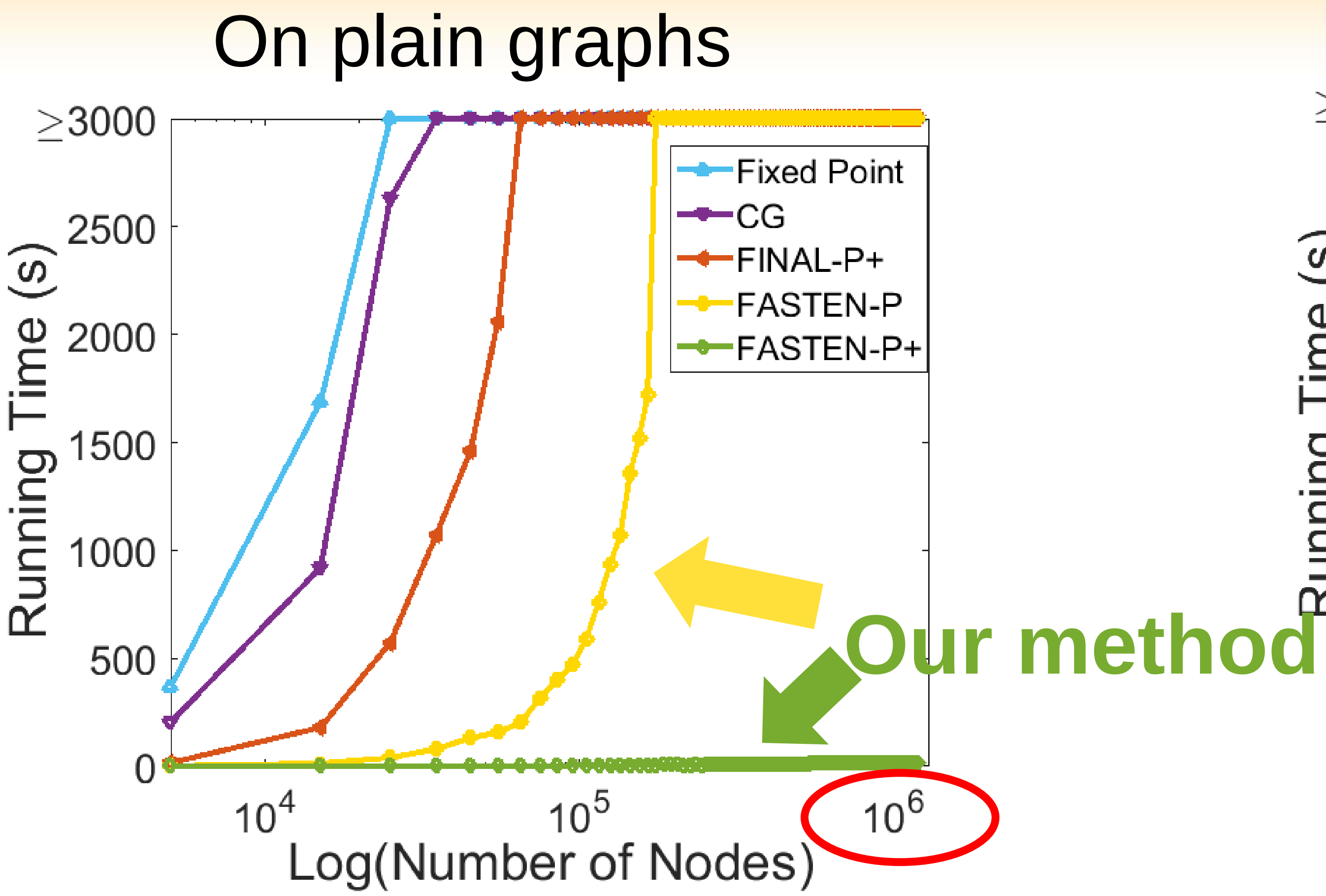
Our method

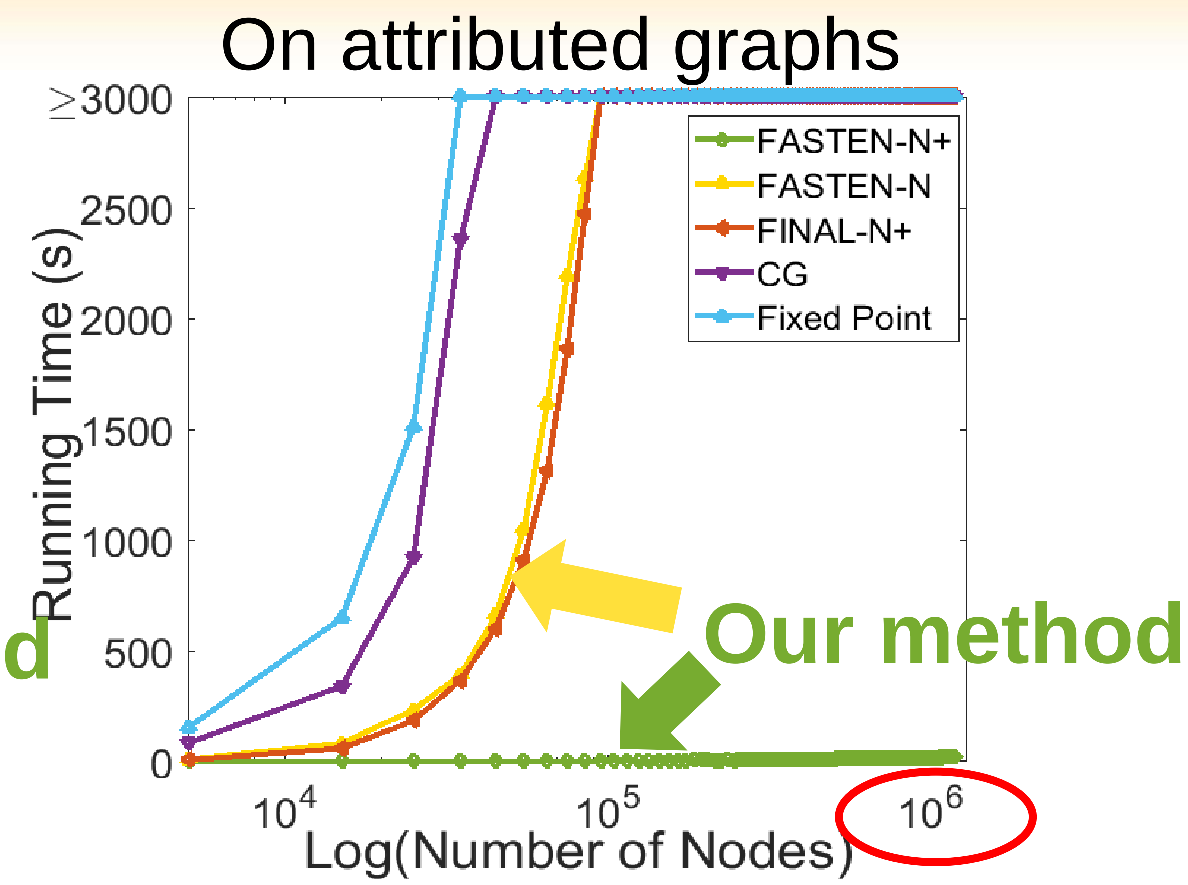


1. DBLP (9,143 nodes)
2. Flickr (12.974 nodes)
3. LastFm (15,436 nodes)
4. Aminer with 25K nodes
5. Aminer with 100K nodes
6. Aminer with 1.2M nodes
7. LinkedIn (6.7M nodes)

- Obs.: maximum speedup: > 10,700 times with 25K-node network.
- Better than approximated methods!

Experimental Result - Scalability

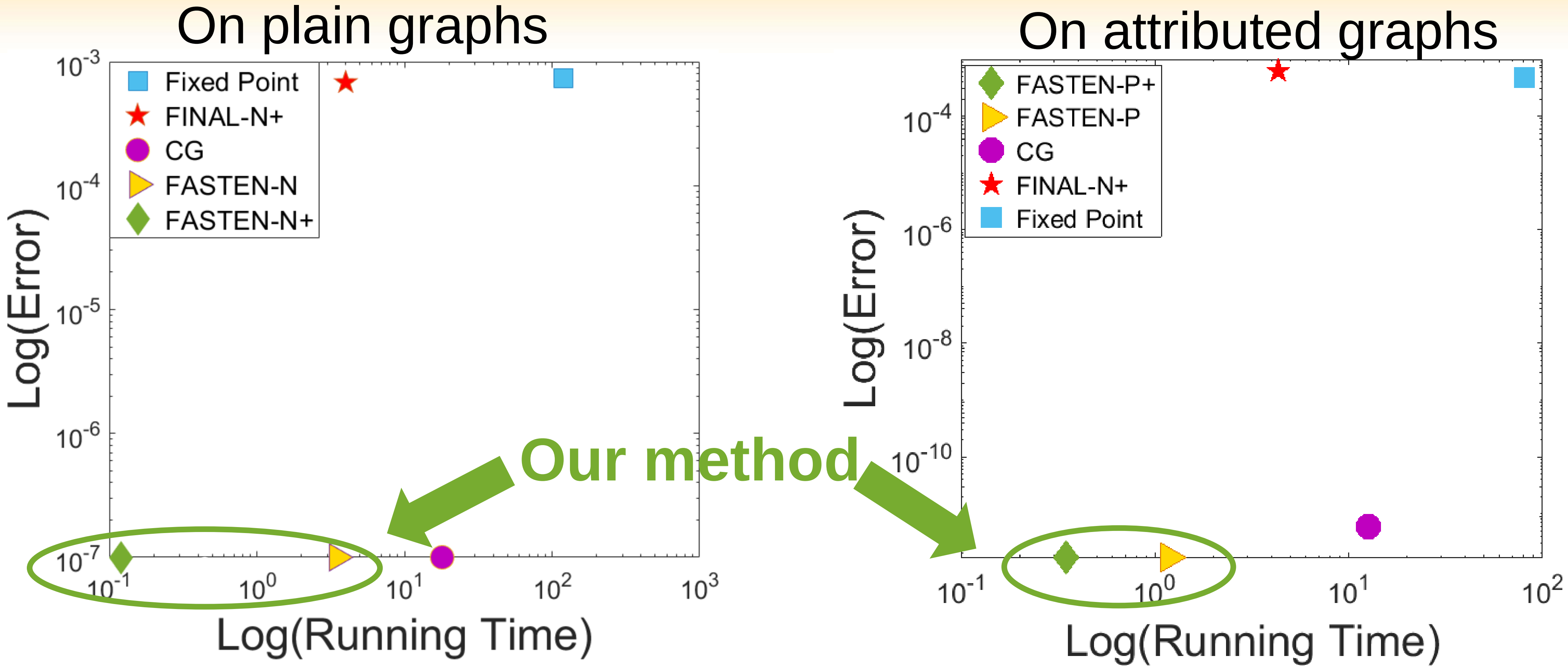




- Obs.: FASTEN-P/N scales almost in accord with FINAL-P+/N+
- FASTEN-P+/N+ scale linearly with regard to # of nodes (to over 1M)

DATA Lab

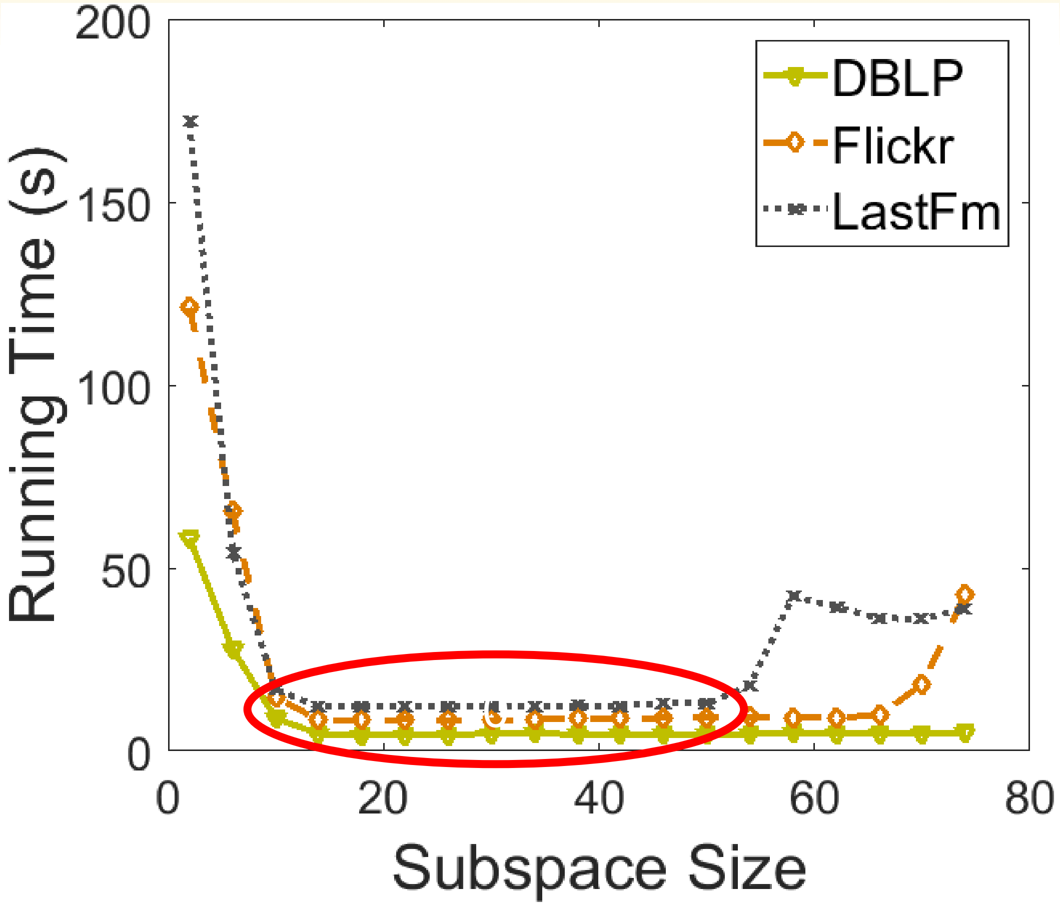
Experimental Result - Effectiveness



- Obs.: FASTEN gives exact solution while having low running time.

Parameter Sensitivity

- e.g.: FASTEN-P:



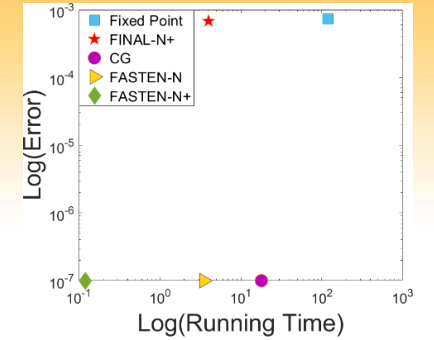
- Obs.: the running time of FASTEN-P stays stable in a range of [14,60].

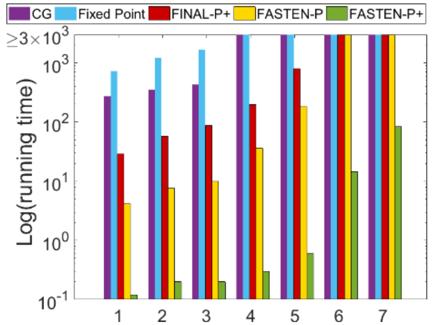
Roadmap

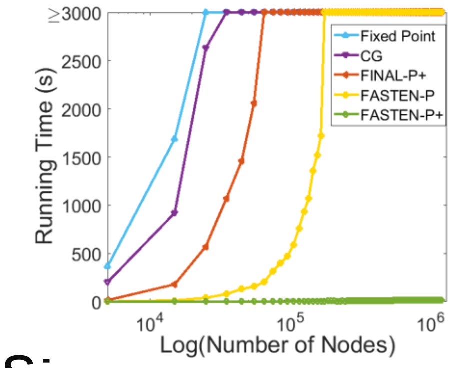
- Motivations ✓
- Background ✓
- Proposed Algorithms for plain graphs ✓
- Proposed Algorithms for attributed graphs ✓
- Experimental Results ✓
- Conclusions

Conclusions

- **Goal**: **Fast** & **exact** solver for (**attributed**) Sylvester equation.
- **Solution**: "FASTEN" family
  - Key idea #1: Generate Kronecker Krylov subspace
  - Key idea #2: Indirect solution representation
  - Key idea #3: Decomposition of Sylvester equation
  - Key idea #4: BCD & FASTEN-P(+) on decomposed equation
- **Results**:
  - Exact solution and *linear* scalability w.r.t the size of input graphs;
  - Significant speedup against traditional methods.







Algorithm	Attributed (Y/N)	Exact Solution (Y/N)	Time Complexity	Space Complexity	
Fixed Point (FP) [Vishwanathan et al' 10]	✓	✓	$O(n^3)$	$O(m^2)$	Traditional methods
Conjugate Gradient (CG) [Y Saad et al' 03]	✓	✓	$O(n^3)$	$O(m^2)$	
Sylv. [Vishwanathan et al' 10]	✓	✓	$O(n^3)$	$O(m^2)$	
ARK [U Kang et al' 12]	✓	✗	$O(n^2)$	$O(n^2)$	Recent methods
Cheetah [L Li et al' 10]	✓	✗	$O(rn^2)$	$O(n^2)$	
NI-Sim [C Li et al' 10]	✗	✗	$O(n^2)$	$O(r^2n^2)$	
FINAL-P [S Zhang et al'16]	✗	✓	$O(mn+n^2)$	$O(n^2)$	
FINAL-NE [S Zhang et al'16]	✓	✓	$O(lmn+ln^2)$	$O(n^2)$	
FINAL-N+ [S Zhang et al'16]	✓	✗	$O(n^2)$	$O(n^2)$	
FASTEN-P	✗	✓	$O(kn^2)$	$O(n^2)$	This Paper
FASTEN-P+	✗	✓	$O(km+k^2n)$	$O(m+kn)$	
FASTEN-N	✓	✓	$O(mn/l+kn^2/l)$	$O(m/l+n^2)$	
FASTEN-N+	✓	✓	$O(km+k^2ln)$	$O(m+kln)$	

DATA Lab

Thank You!