The Forward-Backward Embedding of Directed Graphs

Thomas Bonald, Nathan de Lara, Alexandre Hollocou

November 14, 2018

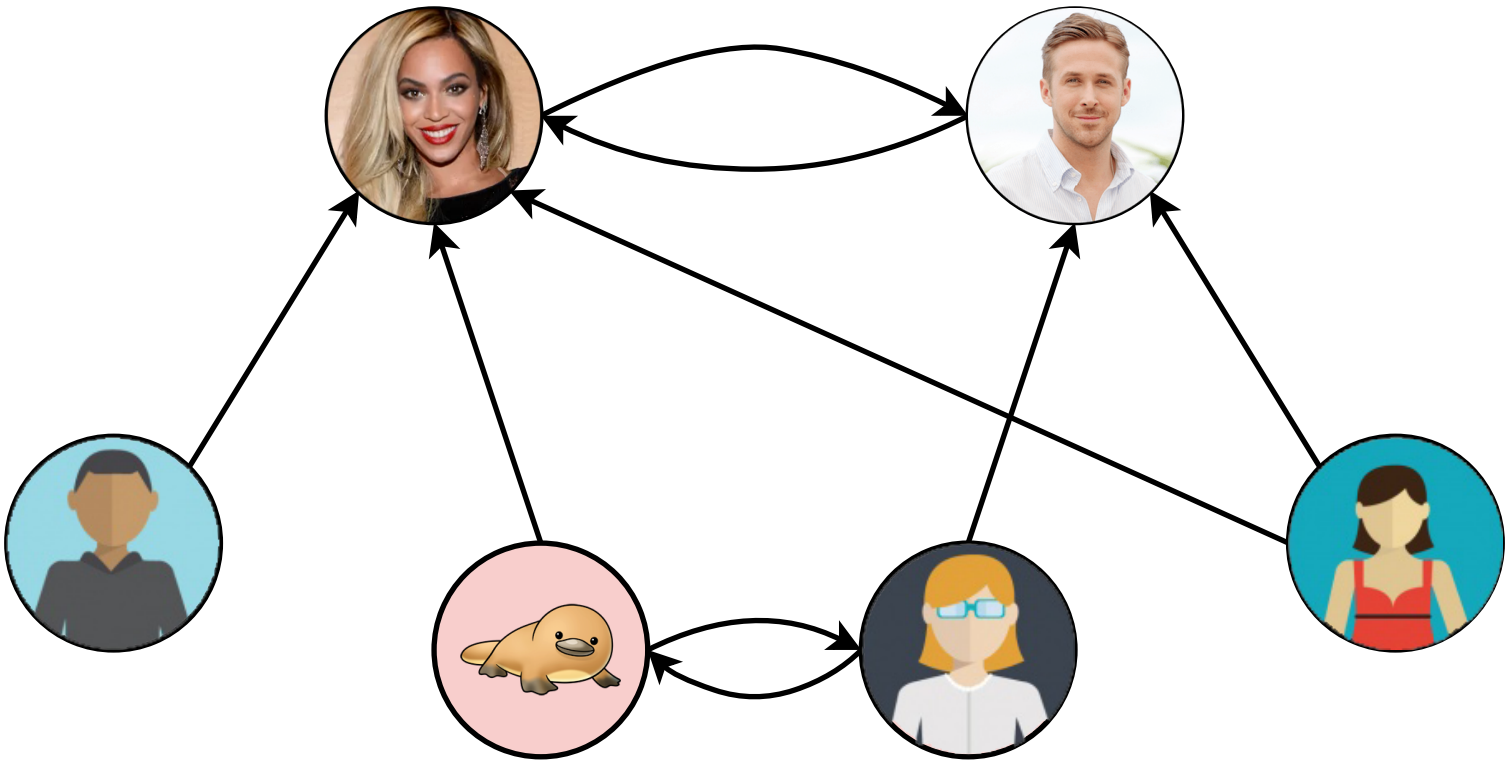
Outline

Section 1

Problem Statement

Directed Graphs

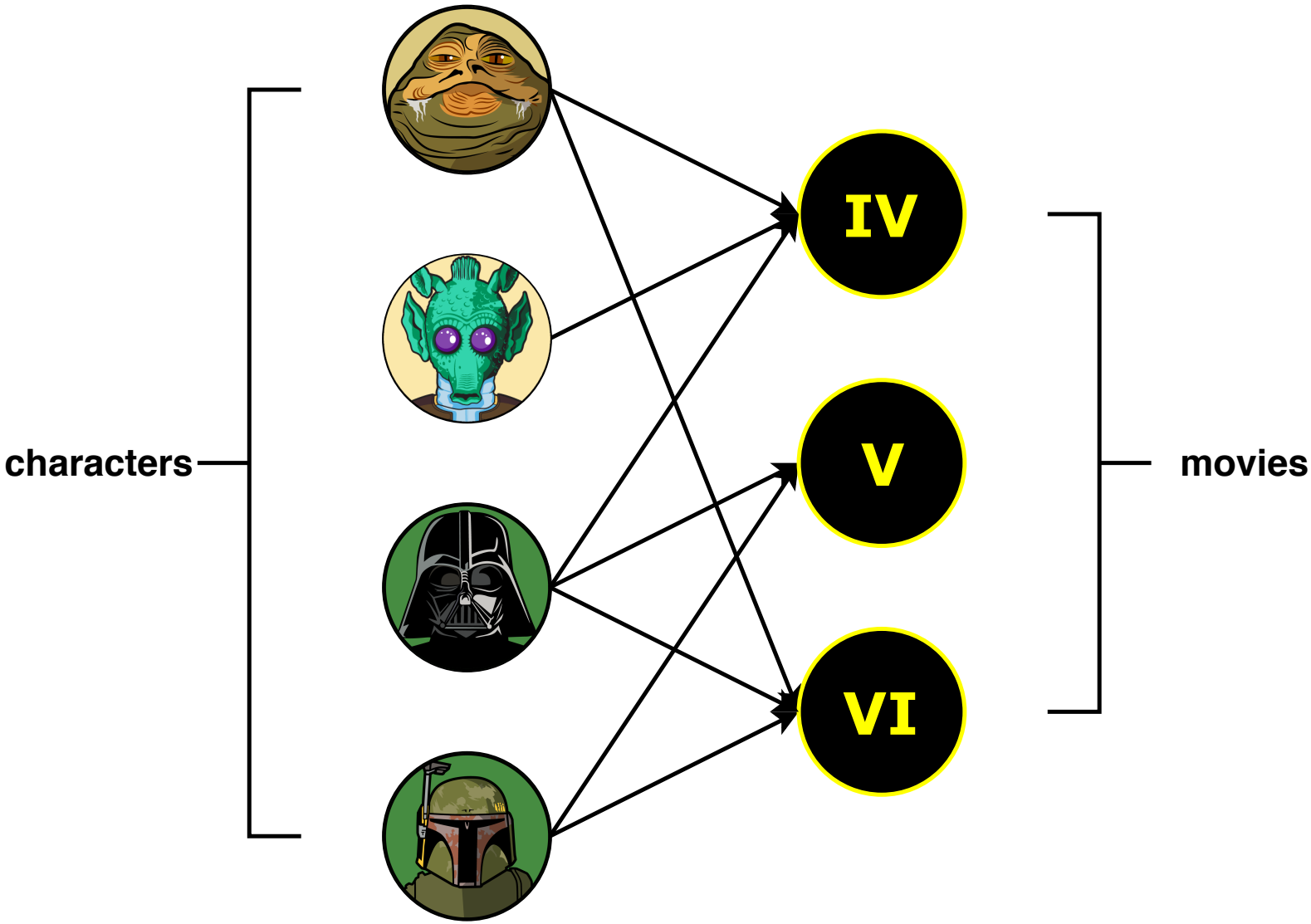
All relationships are not reciprocal...
Consider a "Twitter-like" graph:



Beyoncé has many fans, however, she only follows her friends.

Bipartite Graphs

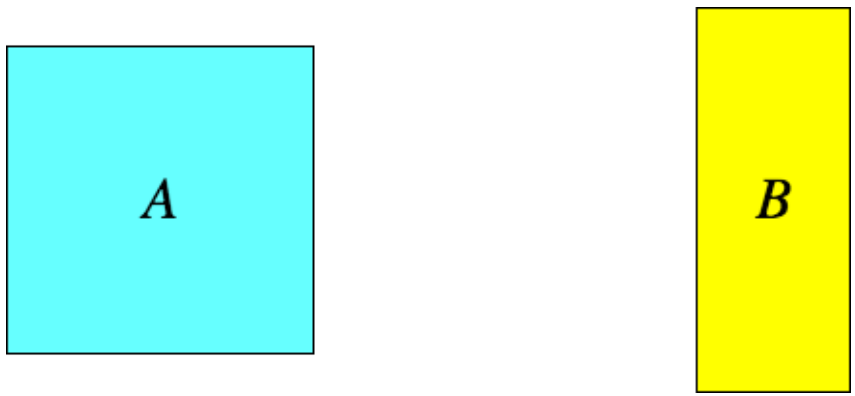
Some relationships are constrained by node categories:



Problem

Let

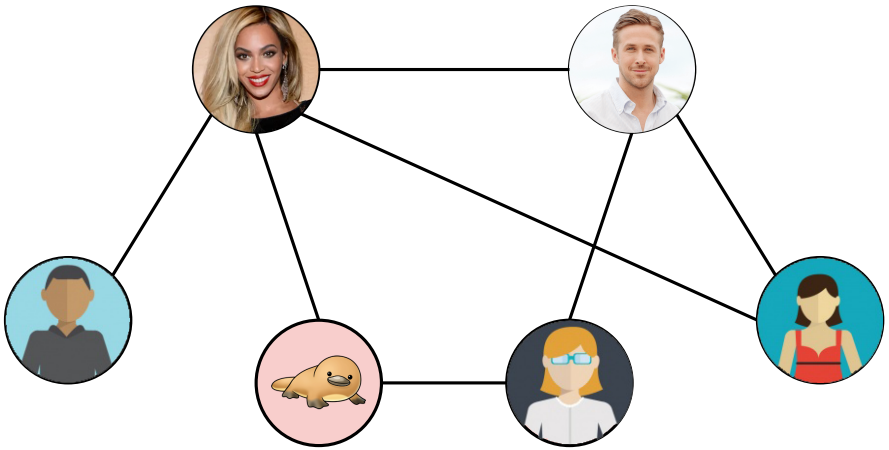
- $A \in \mathbb{R}_+^{|V| \times |V|}$ be the adjacency matrix of a directed graph.
- $B \in \mathbb{R}_+^{|V_1| \times |V_2|}$ be the adjacency matrix of a bipartite graph.

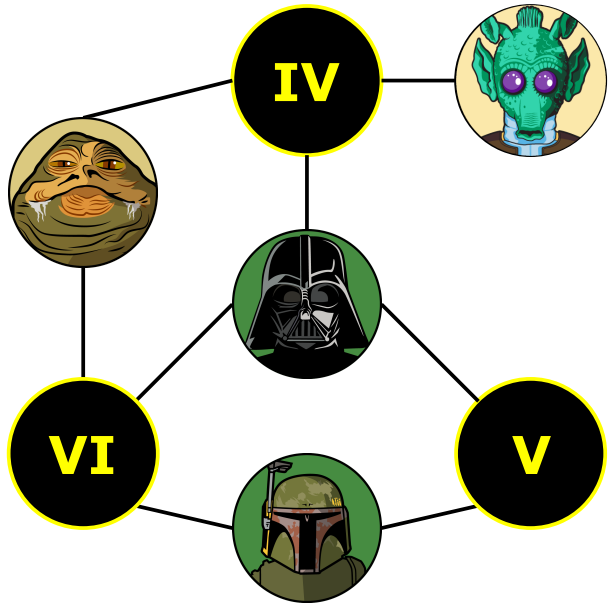


How to leverage the power of spectral embedding given that:

- A is not necessarily symmetric,
- B is not even square ?

Naive solution: lose the specific structure





$A + A^T$

$\max(A, A^T)$

0	B
B^T	0

Less naive solution: the cocitation graph

Consider the undirected graph $\tilde{G}$ where there is an edge between node i and node j if and only if they have a common successor in G.

The participation of each successor to the weight of the edge is normalized by its *in-degree*.

Let D_{in} be the diagonal matrix of *in-degrees*, the adjacency matrices:

- $\tilde{A} = AD_{in}^{-1}A^T$,
- $\tilde{B} = BD_{in}^{-1}B^T$,

are both square and symmetric.

A few properties of the cocitation graph

Why the normalization by D_{in}?

- Beyoncé has 15 millions followers on Twitter, LINCS has 31.
- The connection between two LINCS fans is thus more informative than between two Beyoncé fans.

The degree of a node in $\tilde{G}$ is equal to its *out-degree* in G:

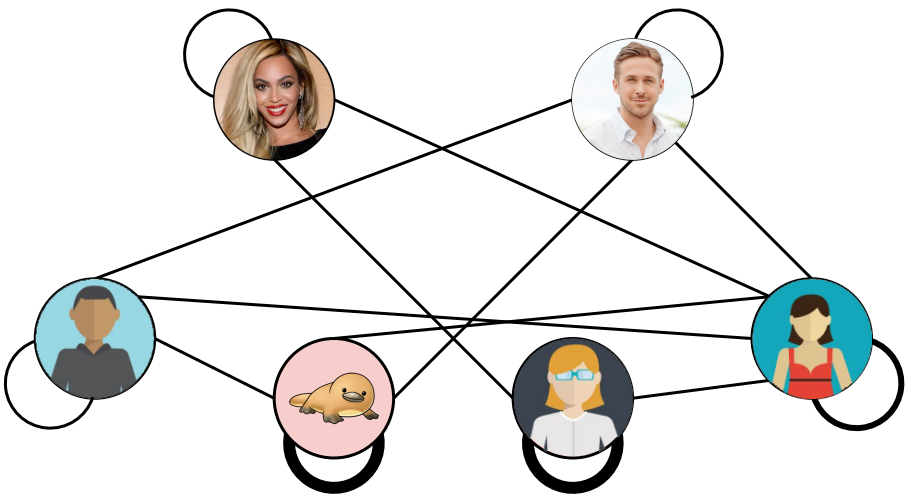
$$\tilde{D} = \tilde{A}\mathbf{1} = AD_{in}^{-1}D_{in}\mathbf{1} = D_{out}.$$

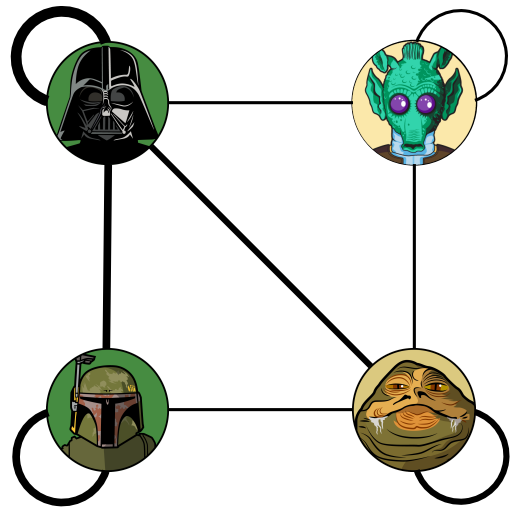
Consequently, the total weight of the graph is also preserved:

$$\tilde{w} = \mathbf{1}^T\tilde{A}\mathbf{1} = \mathbf{1}^T D_{out} = w.$$

However...

A node with *in-degree* d in G generates a clique of size d in $\tilde{G}$ and thus $d(d-1)/2$ edges.





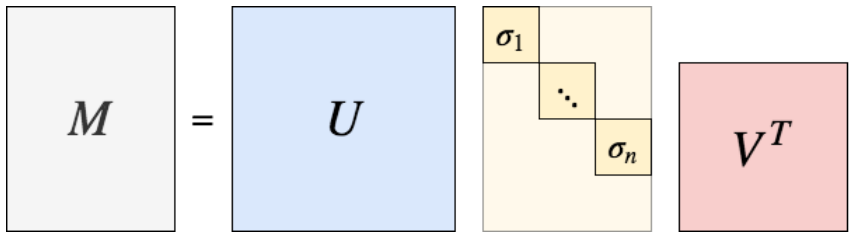
Beyoncé alone generates a clique with 10^{14} edges. Not convenient to store in memory...

Section 2

Proposed Solution

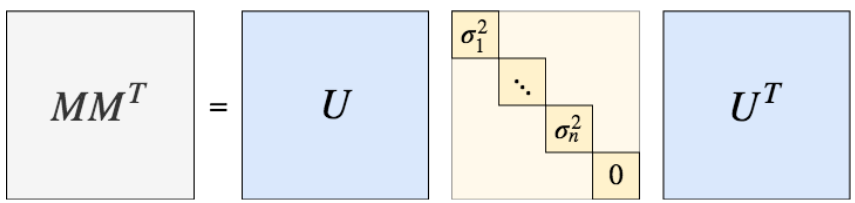
A quick refresher on linear algebra

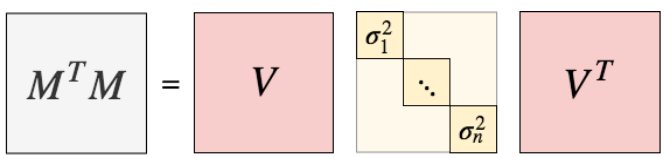
The Singular Value Decomposition (SVD)



with U, V orthonormal matrices: $UU^T = I$ and $VV^T = I$.

Connected to the eigenvalue decomposition through:





Normalized adjacencies

Let

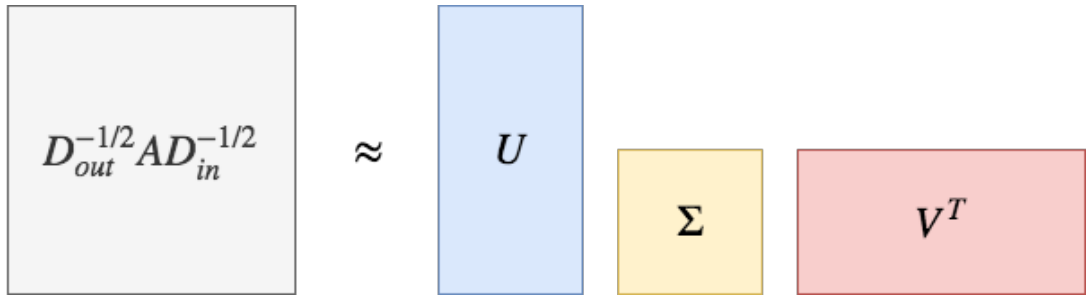
- $\bar{A} = D_{out}^{-1/2} A D_{in}^{-1/2}$ be the normalized adjacency of A,
- $\bar{B} = D_1^{-1/2} B D_2^{-1/2}$ be the normalized adjacency of B,
- $\bar{\tilde{A}} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$ be the normalized adjacency of $\tilde{A}$,
- $\bar{\tilde{B}}$ be the normalized adjacency of $\tilde{B}$.

These matrices are connected through:

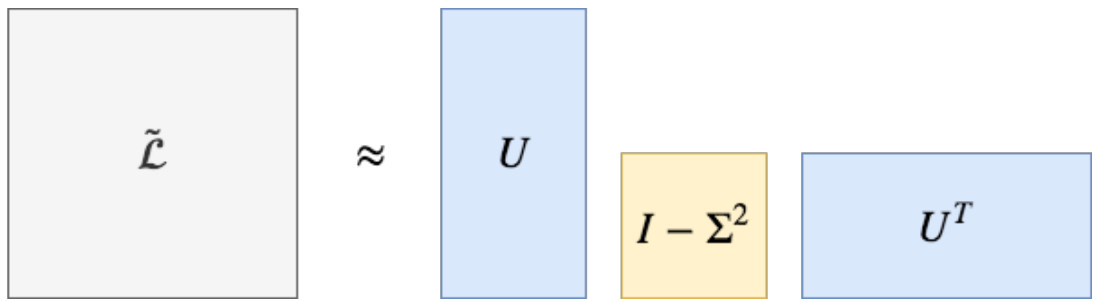
- $\bar{\tilde{A}} = \bar{A}\bar{A}^T$,
- $\bar{\tilde{B}} = \bar{B}\bar{B}^T$.

Our solution: an implicit decomposition

Consider the SVD of $\bar{A}$:



The normalized Laplacian of $\tilde{G}$, $\tilde{L} = I - \bar{\tilde{A}}$ satisfies:



Thus, decomposing $\tilde{L}$ **does not require** to compute $\tilde{G}$.

The embedding

Let $\Gamma = (I - \Sigma^2)^{1/2}$, we define the embedding:

$$X = D_{out}^{-1/2} U \Gamma^{\dagger}.$$

The i^{th} row of X represents the embedding of node i.

We call it the *forward embedding* as opposed to the *backward embedding*:

$$Y = D_{in}^{-1/2} V \Gamma^{\dagger}.$$

Truncate to the k largest singular values to get a low dimensional (bust fast) embedding.

The Forward-Backward random walk

Consider a random walk in G where edges are followed in forward and backward directions alternatively:

1. a successor k of i is chosen uniformly at random,
2. a predecessor j of k is chosen uniformly at random.

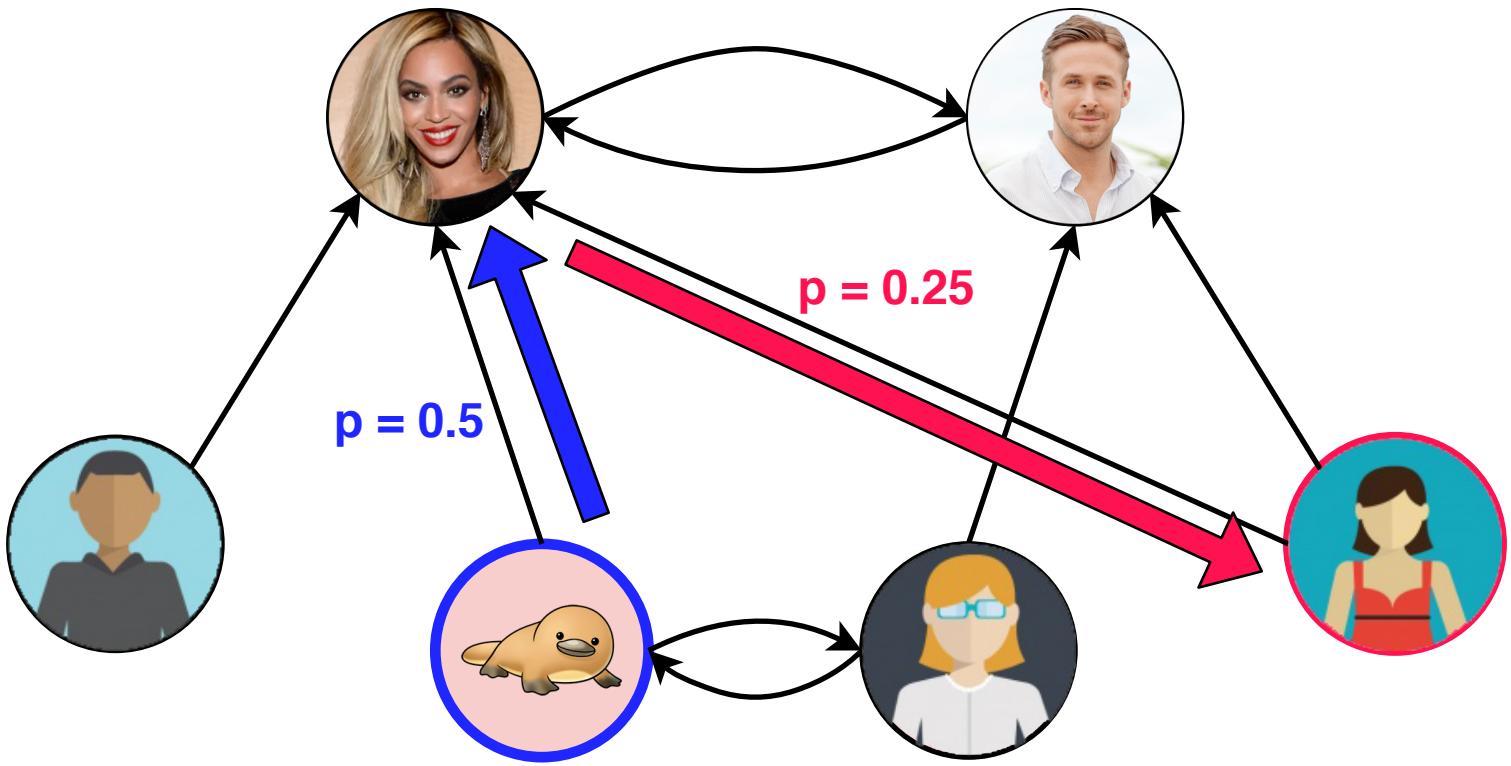


Figure 1: **Blue:** forward move. **Red:** backward move.

Interpreting the embedding

The transition matrix of the Markov chain associated to the forward-backward walk

$$\tilde{P} = D_{out}^{-1} A D_{in}^{-1} A^T = \tilde{D}^{-1} \tilde{A},$$

is the transition matrix of a standard random walk on $\tilde{G}$.

Hence, the embedding of G has all the properties of a spectral embedding of $\tilde{G}$:

- Hitting time: $H_{ij} \propto (X_i - X_j)^T X_j$.
- Commute time: $C_{ij} \propto ||X_i - X_j||^2$.
- Physical interpretations...

The average of who you like is the average of who likes you

The Forward and Backward embeddings are connected through:

- $D_{out}^{-1} AY = X\Sigma$,
- $D_{in}^{-1} AX = Y\Sigma$.

Σ acts as a scaling factor so that:

- Beyoncé as a star is defined by the average of its followers.
- Beyoncé as a regular Twitter user is defined by the average of the accounts she follows.

Section 3

Experiments

Node clustering

Cluster the graph of hyperlinks between pages of *Wikipedia for schools*:

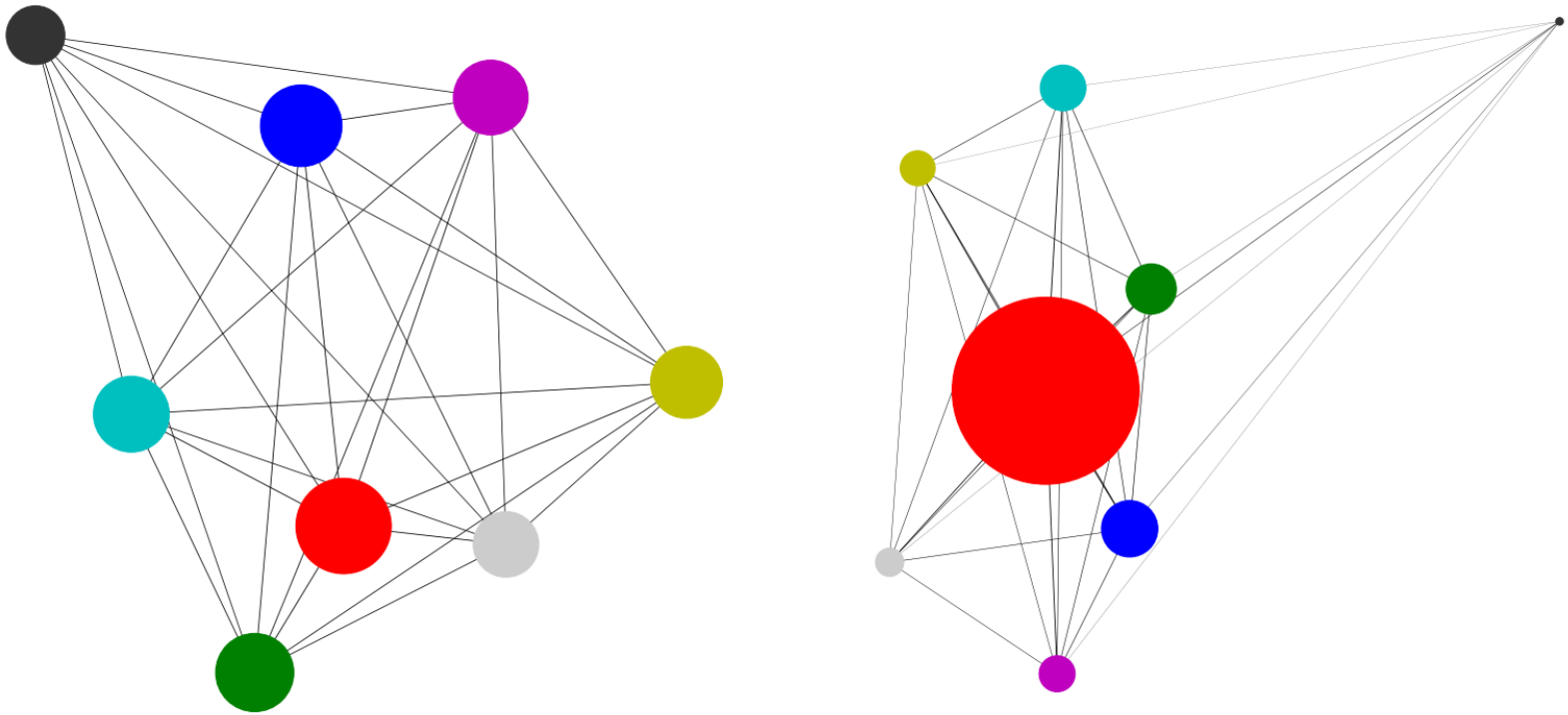
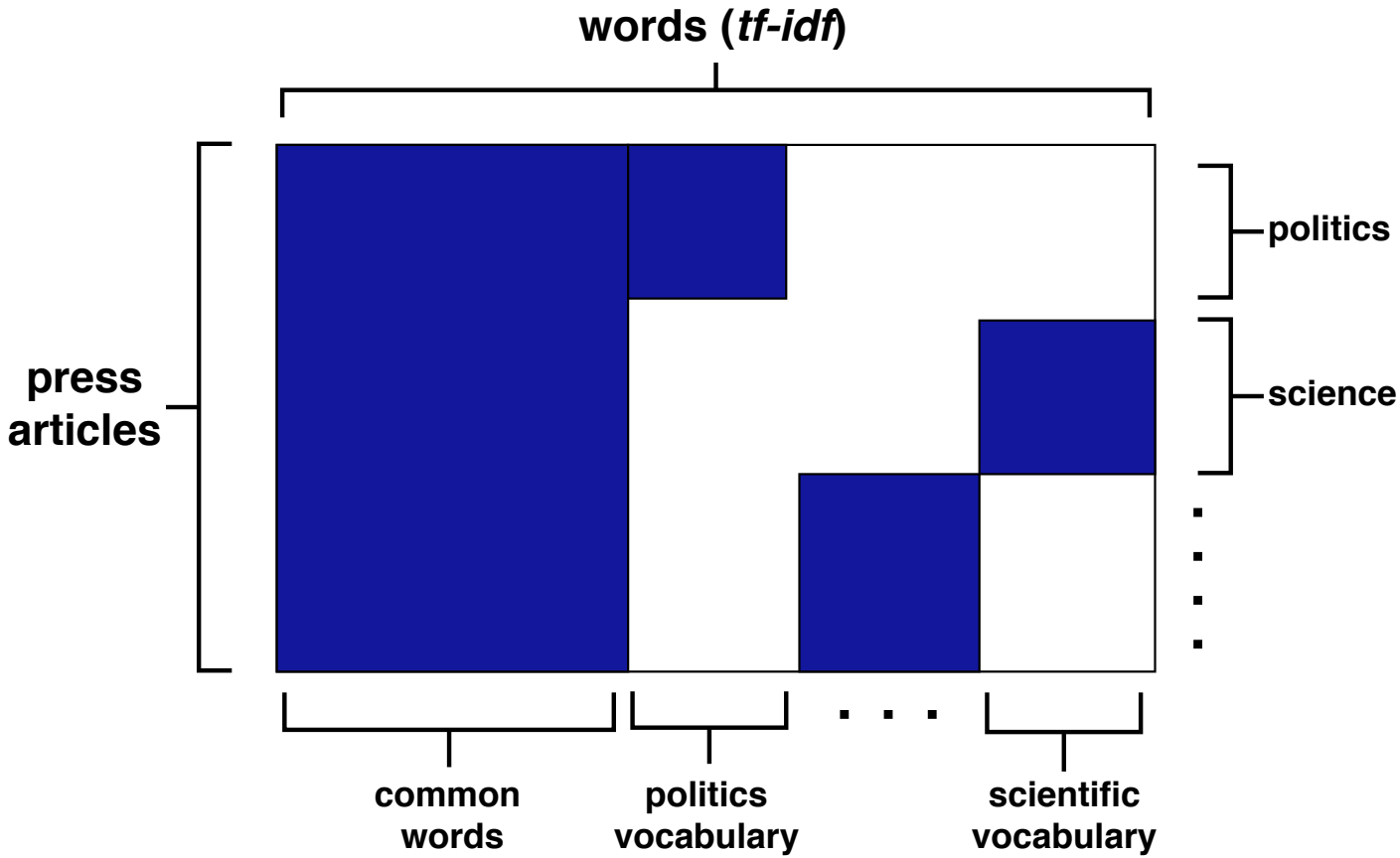


Figure 2: **Left:** Forward-Backward + cosine. **Right:** Laplacian Eigenmaps + L2.

Co-clustering case: the *20newsgroup* dataset



	Id	Dh	LE	FB
v score	0.33	0.44	TO	**0.64**

Figure 3: **Top:** *article-word* bipartite adjacency matrix.
Bottom: v measure score for different embeddings + KMeans.

Scalability

There are highly scalable randomized methods to compute the SVD.

Experiments on Hyperion cluster at LINCS with real datasets from KONECT[1]:

graph	# edges	time	graph	# edges	time
LI	17 359 346	19s	FG	8 545 307	6s
HUr	18 854 882	42s	M3	10 000 054	4s
PL	30 622 564	56s	DI	14 414 659	16s
FL	33 140 017	47s	Ls	19 150 868	8s
LJ	68 475 391	119s	RE	60 569 726	22s

Table 1: Running times. **Left:** directed graphs. **Right:** bipartite graphs.

[1] http://konect.uni-koblenz.de/

What if the graph is undirected?

If the graph is not directed, the eigenvectors of the standard normalized Laplacian are the same as for the Laplacian of the cocitation graph.

However, the two embeddings differ by the normalization:

- normalization by $\sqrt{\lambda_k}$ in the first case,
- normalization by $\sqrt{\lambda_k(2-\lambda_k)}$ in the latter.

Recall that:

- $\lambda_k = 0 \iff$ connected component,
- $\lambda_k = 2 \iff$ bipartite structure.

Conclusion

What is to take away:

- Do not bother building a cocitation graph.
- Proper normalization gives interpretable results.
- Use cosine similarity instead of euclidean distance.
- SVD is highly scalable.

Scikit-network

Thank you for your attention.

The Forward-Backward embedding and many other algorithms will soon be available in **scikit-network**:
https://github.com/sknetwork-team/scikit-network

Try it with:
pip install scikit-network