Frobenius Matrix is Awesome!

Piotr Sankowski, **Karol Węgrzycki**

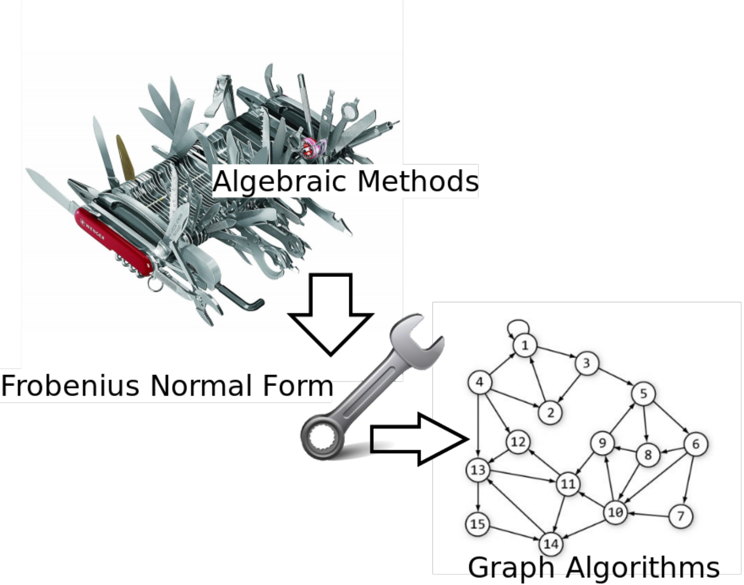
University of Warsaw

Paper: bit.ly/frobenius-form

Highlights of Algorithms 2017

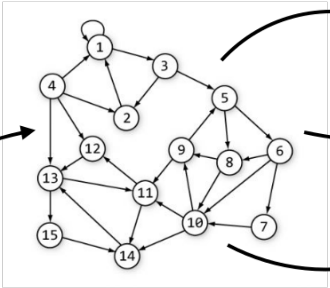
European Research Council
Established by the European Commission

Overview


Algebraic Methods
Frobenius Normal Form
Graph Algorithms

Overview

Frobenius Normal Form



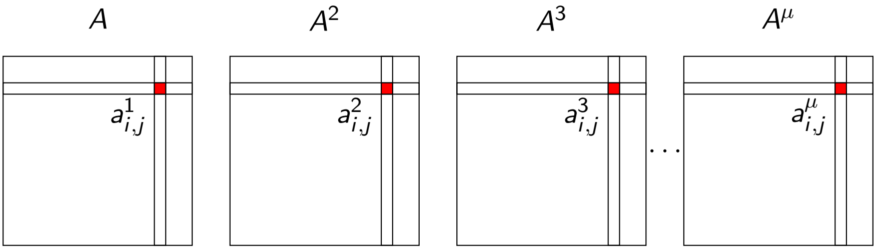
Determining cycles

Counting Walks

Distance Queries

Our observation

We get a matrix A on input. For a query (i, j) we return μ numbers

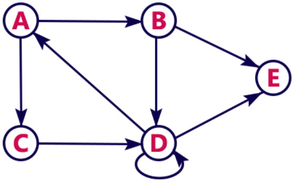


- Preprocessing: $\tilde{O}(n^{2.38})$
- Query time: $O(n \log n)$

Naive in $O(\mu n^{2.38})$ preprocessing time and $O(1)$ query.

Relationship with Graphs

Incidence Matrix for **directed** graphs:



	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	1	1
C	0	0	0	1	0
D	1	0	0	1	1
E	0	0	0	0	0

$$a_{i,j} = \begin{cases} 1 & \text{if } i \to j \in E(G) \\ 0 & \text{if } i \to j \notin E(G) \end{cases}$$

Example All Pairs All Walks

Problem All Pairs All Walks

Return an array A, such that for every pair of vertices $u, v \in G$ and every $k \in \{1, \ldots, D\}$ an element $A[u, v, k]$ is the **number of distinct walks** from u to v of length k.

Example All Pairs All Walks

Problem All Pairs All Walks

Return an array A, such that for every pair of vertices $u, v \in G$ and every $k \in \{1, \ldots, D\}$ an element $A[u, v, k]$ is the **number of distinct walks** from u to v of length k.

Naive

$O(Dn^{\omega})$ time where $D = O(n)$.
Compute $A, A^2, A^3, \ldots, A^D$.
Bad case $O(n^{3.28})$ time.

Example All Pairs All Walks

Problem All Pairs All Walks

Return an array A, such that for every pair of vertices $u, v \in G$ and every $k \in \{1, \dots, D\}$ an element $A[u, v, k]$ is the **number of distinct walks** from u to v of length k.

Naive

$O(Dn^{\omega})$ time where $D = O(n)$.
Compute $A, A^2, A^3, \dots, A^D$.
Bad case $O(n^{3.28})$ time.

By using our theorem

$O(n^3 \log n)$ time.
For every pair ($O(n^2)$ pairs) we can compute $(A^1)_{i,j}, (A^2)_{i,j}, (A_3)_{i,j}, \dots, (A^D)_{i,j}$ in time $O(n \log n)$

All Nodes Shortest Cycles

All Nodes Shortest Cycles

For every vertex in the graph return the length of the shortest cycle that contains it.

Date	Authors	Time	Comments
–	Naive	$O(n^3)$	–
2011	Yuster	$\tilde{O}(n^{(\omega+3)/2})$	Undirected
Nov 2017	Agarwal and Ramachandran	$\tilde{O}(n^{\omega})$	Undirected
Nov 2017	**This Paper**	$\tilde{O}(n^{\omega})$	**Directed**

Our Technique

Compute $(A^1)_{i,i}, (A^2)_{i,i}, \ldots, (A^{O(D)})_{i,i}$ for all vertices.
Time: $\tilde{O}(n^{\omega} + n^2 \log n)$.

Sets on cycles

Sets on cycles

Determine the set of vertices $S(t)$ that lie on some cycle of length at most t.

Date	Authors	Time	Comments
2011	Yuster	$\tilde{O}(tn^{\omega})$	Returns: $S(t)$
2015	Cygan et al.	$\tilde{O}(n^{\omega})$	Returns: $S(t)$
2017	**This Paper**	$\tilde{O}(n^{\omega})$	Returns: $S(1), \ldots, S(D)$

Our technique

Same as in All Nodes Shortest Cycles.

Distance Queries

Distance Queries

Preprocess graph in such a way, that you can answer queries about distance $\delta(u, v)$ fast.

Authors	**Preprocessing**	**Query**	Comments
Naive APSP	$O(n^{2.52})$	$O(1)$	
Yuster Zwick	$O(n^{2.38})$	$O(n)$	$\delta(u, v)$
This Paper	$\tilde{O}(n^{2.38})$	$O(n \log n)$	$\delta^1(u, v), \ldots \delta^D(u, v)$

Frobenius Normal Form

Frobenius Matrix

$$A = UFU^{-1} = U \begin{bmatrix} C_1 & & & & 0 \\ & C_2 & & & \\ & & C_3 & & \\ & & & \ddots & \\ 0 & & & & C_k \end{bmatrix} U^{-1}.$$

Frobenius Normal Form

Companion Matrix

$$C_i = \begin{bmatrix} 0 & & \cdots & & 0 & -c_0 \\ 1 & 0 & & & 0 & -c_1 \\ & 1 & \ddots & & \vdots & -c_2 \\ & & \ddots & 0 & & \vdots \\ & & & 1 & 0 & -c_{r-2} \\ 0 & & & & 1 & -c_{r-1} \end{bmatrix} \in K^{r \times r}.$$

Example

Powering companion Matrix

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 5 \end{bmatrix},$$

Example

Powering companion Matrix

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 5 \end{bmatrix}, \ C^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 2 & 11 \\ 1 & 0 & 0 & 3 & 17 \\ 0 & 1 & 0 & 4 & 23 \\ 0 & 0 & 1 & 5 & 29 \end{bmatrix},$$

Example

Powering companion Matrix

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 5 \end{bmatrix}, \; C^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 2 & 11 \\ 1 & 0 & 0 & 3 & 17 \\ 0 & 1 & 0 & 4 & 23 \\ 0 & 0 & 1 & 5 & 29 \end{bmatrix}, \; C^3 \begin{bmatrix} 0 & 0 & 1 & 5 & 29 \\ 0 & 0 & 2 & 11 & 63 \\ 0 & 0 & 3 & 17 & 98 \\ 1 & 0 & 4 & 23 & 133 \\ 0 & 1 & 5 & 29 & 168 \end{bmatrix},$$

Example

Powering companion Matrix

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 5 \end{bmatrix}, \; C^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 2 & 11 \\ 1 & 0 & 0 & 3 & 17 \\ 0 & 1 & 0 & 4 & 23 \\ 0 & 0 & 1 & 5 & 29 \end{bmatrix}, \; C^3 \begin{bmatrix} 0 & 0 & 1 & 5 & 29 \\ 0 & 0 & 2 & 11 & 63 \\ 0 & 0 & 3 & 17 & 98 \\ 1 & 0 & 4 & 23 & 133 \\ 0 & 1 & 5 & 29 & 168 \end{bmatrix},$$

$$C^4 = \begin{bmatrix} 0 & 1 & 5 & 29 & 168 \\ 0 & 2 & 11 & 63 & 365 \\ 0 & 3 & 17 & 98 & 567 \\ 0 & 4 & 23 & 13 & 770 \\ 1 & 5 & 29 & 16 & 973 \end{bmatrix},$$

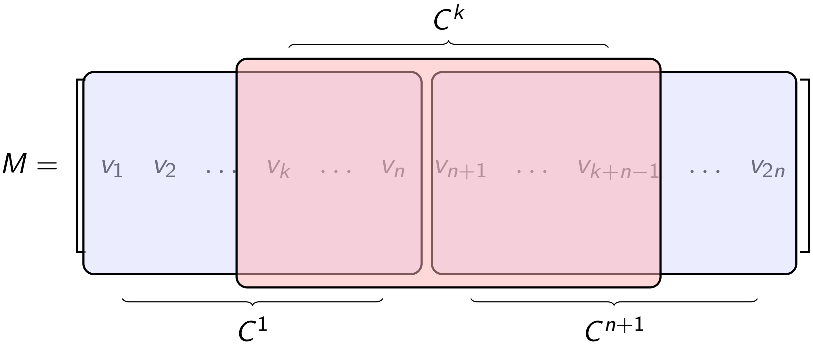
Example

Powering companion Matrix

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 5 \end{bmatrix}, \ C^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 2 & 11 \\ 1 & 0 & 0 & 3 & 17 \\ 0 & 1 & 0 & 4 & 23 \\ 0 & 0 & 1 & 5 & 29 \end{bmatrix}, \ C^3 \begin{bmatrix} 0 & 0 & 1 & 5 & 29 \\ 0 & 0 & 2 & 11 & 63 \\ 0 & 0 & 3 & 17 & 98 \\ 1 & 0 & 4 & 23 & 133 \\ 0 & 1 & 5 & 29 & 168 \end{bmatrix},$$

$$C^4 = \begin{bmatrix} 0 & 1 & 5 & 29 & 168 \\ 0 & 2 & 11 & 63 & 365 \\ 0 & 3 & 17 & 98 & 567 \\ 0 & 4 & 23 & 13 & 770 \\ 1 & 5 & 29 & 16 & 973 \end{bmatrix}, \ C^5 = \begin{bmatrix} 1 & 5 & 29 & 168 & 973 \\ 2 & 11 & 63 & 365 & 2114 \\ 3 & 17 & 98 & 567 & 3284 \\ 4 & 23 & 133 & 770 & 4459 \\ 5 & 29 & 168 & 973 & 5635 \end{bmatrix}$$

Frobenius Normal Form



Future Work

Dynamic APSP

[SODA 2017] Dynamic APSP in $O(n^{2.5})$ worst case update time (for directed unweighted graphs)

- Can we get $O(kn^2)$ for graph parameter $k = O(n^{0.5})$?
- $O(n^{2.49})$ **worst case** update time algorithm is a major open problem.

Random Walker on a graph

$$P_{i,j} = \begin{cases} \frac{1}{\text{deg\_out}(i)} & \text{if } i \to j \\ 0 & \text{otherwise} \end{cases}$$

$P^t_{i,j}$ is the probability of getting from i to j in t steps.

Thank You!

Arxiv link

`http://bit.ly/frobenius-form`