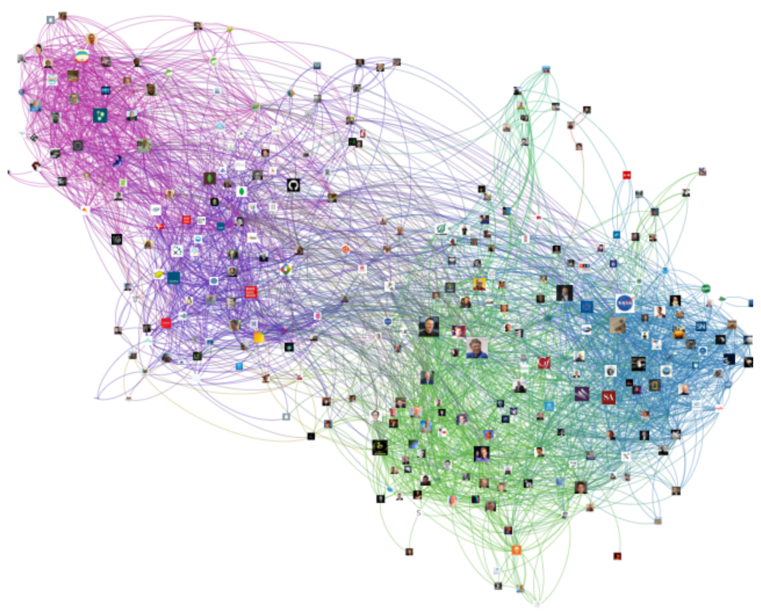
Large Scale Density-friendly Graph Decomposition via Convex Programming

Maximilien Danisch, T-H. Hubert Chan and Mauro Sozio

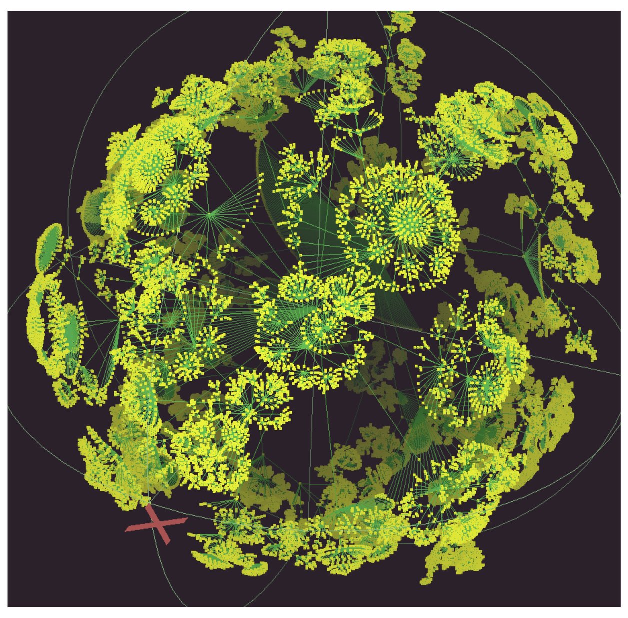
LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France
Department of Computer Science, The University of Hong Kong, Hong Kong, China

Motivation: mining large graphs

Twitter



Internet



The density-friendly decomposition (Tatti and Gionis WWW2015) is interesting as it **merges two classic graph mining concepts**: **(i)** the k-core decomposition and **(ii)** the notion of dense subgraph.

Contributions

Density-friendly is now as intuitive and almost as fast to compute as PageRank.

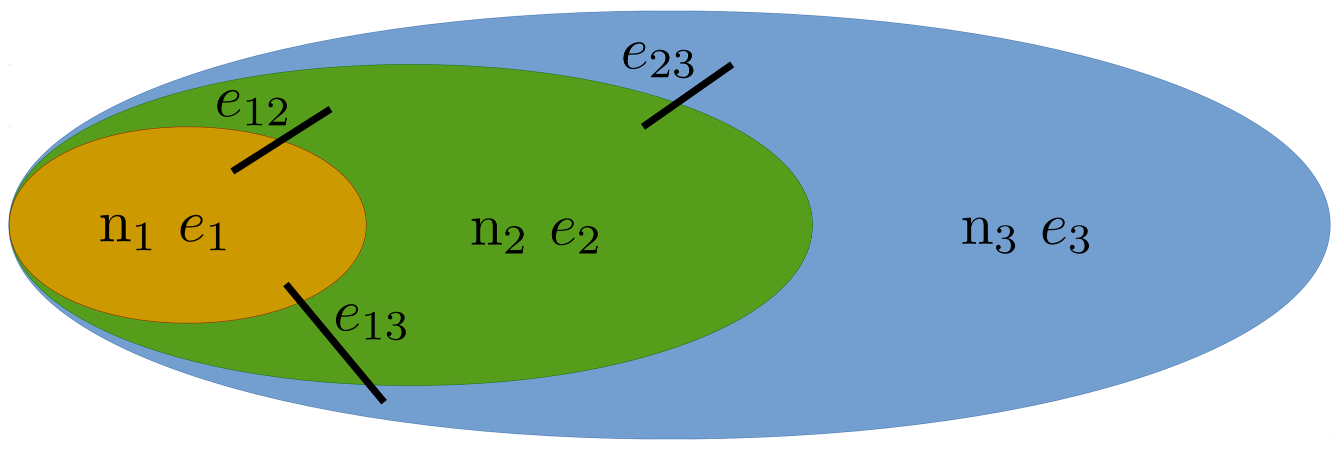
- We gave a new and **intuitive definition** of the density-friendly decomposition.
- We made an interesting link between the density-friendly decomposition and **convex programing**.
- We **scaled up** the computation of the density-friendly decomposition using the **Frank-Wolfe algorithm**.

Full paper at **WWW2017**. Code in C: `https://github.com/maxdan94`

Definition

Collection of non-overlapping sets of nodes $\{B_i\}$, such that

- B_1 maximizes $\frac{e_1}{n_1}$ and has maximum size (B_1 is the densest subgraph),
- B_2 maximizes $\frac{e_2+e_{12}}{n_2}$ and has maximum size,
- B_3 maximizes $\frac{e_3+e_{13}+e_{23}}{n_3}$ and has maximum size, ...

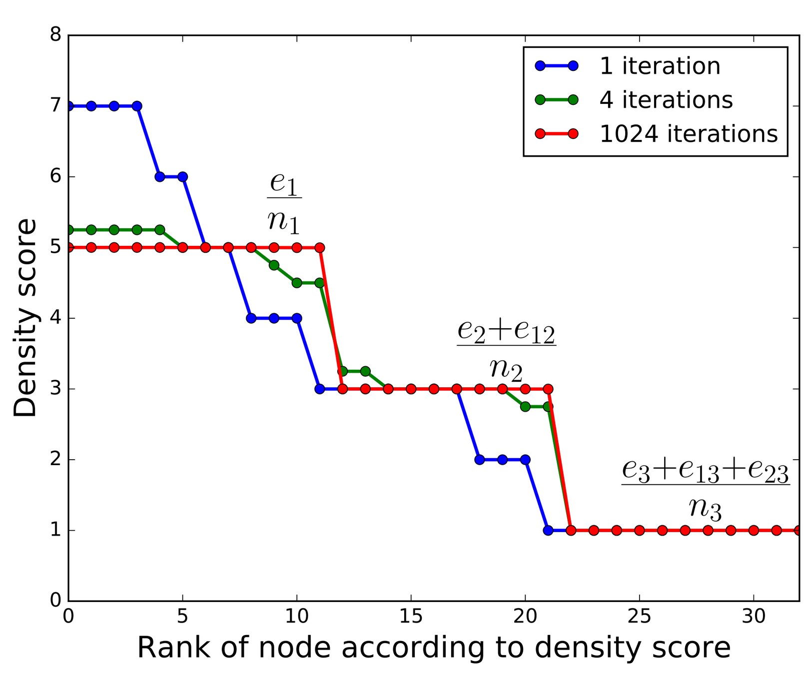


Theorem: our definition is equivalent to the one of Tatti and Gionis WWW2015.

A very simple, yet powerful, algorithm

Algorithm 1 Compute density score

```
function MKSCORE(G,t)
    Set r[i] = 0 for each node i in G
    iterate t times
        for each edge (i,j) in G do
            if r[i] ≤ r[j] then
                r[i]++
            else
                r[j]++
    return r/t
```



Theorem: rank the nodes in decreasing order of density score:

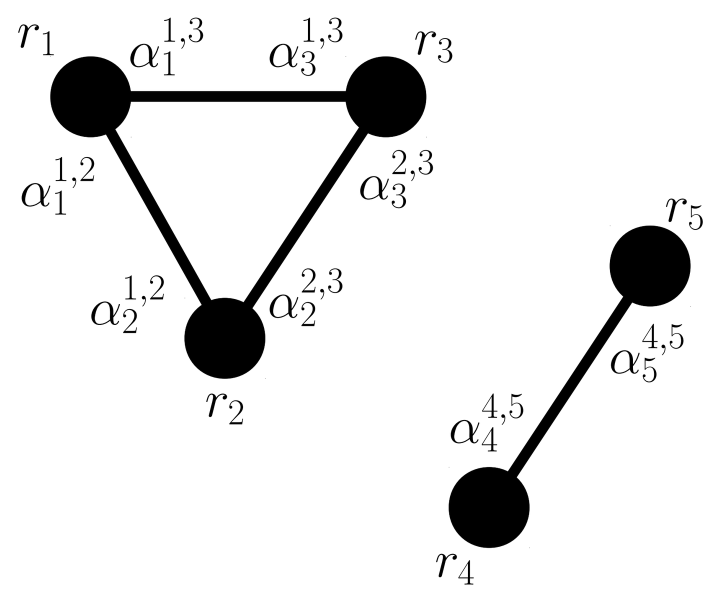
- for t large enough, the nodes in the densest subgraph are ranked first;
- for t large enough, we have a density-friendly ordering.

Theoretical analysis via convex programing

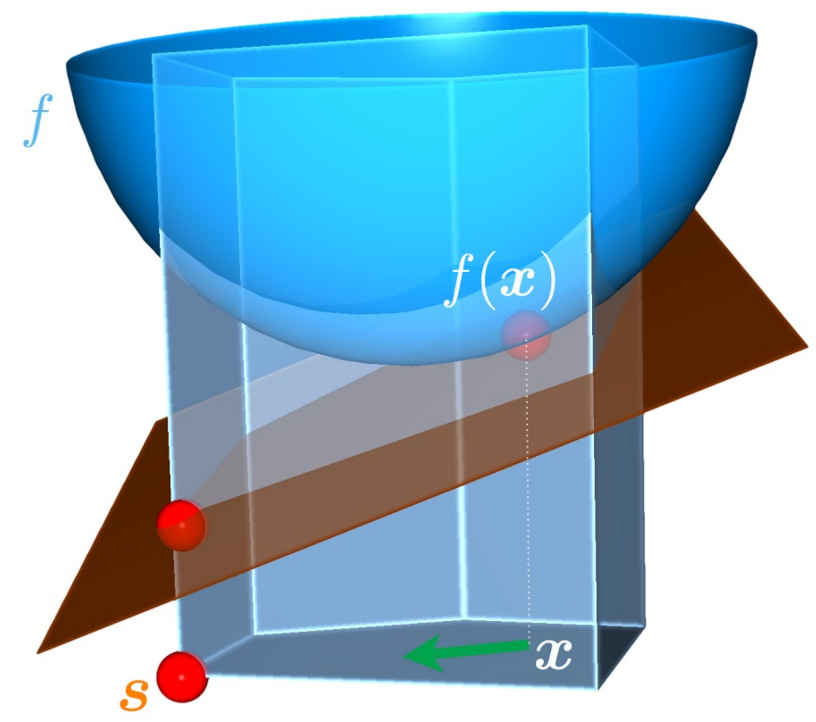
Given an edge-weighted (hyper) graph $G = (V, E, w)$, we consider the following quadratic convex programing.

$$CP(G)$$

$$\begin{aligned} \min \quad & \sum_{u \in V} r_u^2 \\ \text{s.t.} \quad & \forall u \in V,\ r_u = \sum_{e: u \in e} \alpha_u^e \\ & \forall e \in E,\ \sum_{u \in e} \alpha_u^e = w_e \\ & \forall u \in e \in E,\ \alpha_u^e \geq 0 \end{aligned}$$



The Frank-Wolfe algorithm is a **projection free gradient-descent** method which has **convergence guarantees** for convex problems.



Applying Frank-Wolfe on our quadratic convex programing leads to an algorithm very similar to our very simple algorithm.

Theorem (correctness)**:** The level sets of an optimal solution r^G to the convex programing give the density-friendly decomposition (r^G gives the densities).
Theorem (running time): After $t > \frac{4\Delta|E|}{\epsilon^2}$ iterations we have $||r^{(t)} - r^G||_2 \leq \epsilon$.

Experiments

Table 1: Our set of large graphs.

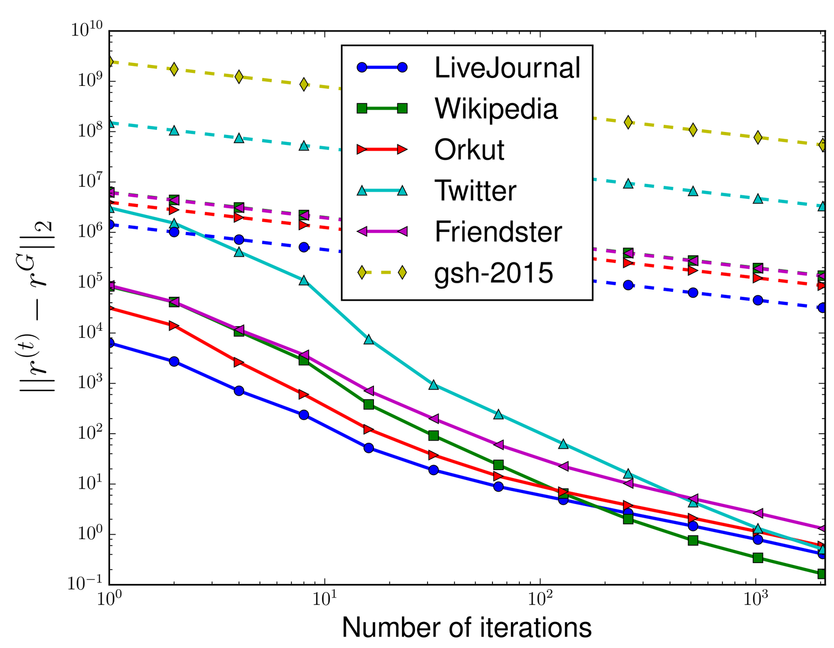
networks	n	m
LiveJournal	4,036,538	34,681,189
Wikipedia	2,080,370	42,336,692
Orkut	3,072,627	117,185,083
Twitter	52,579,683	1,614,106,500
Friendster	124,836,180	1,806,067,135
gsh-2015	988,490,691	25,690,705,119

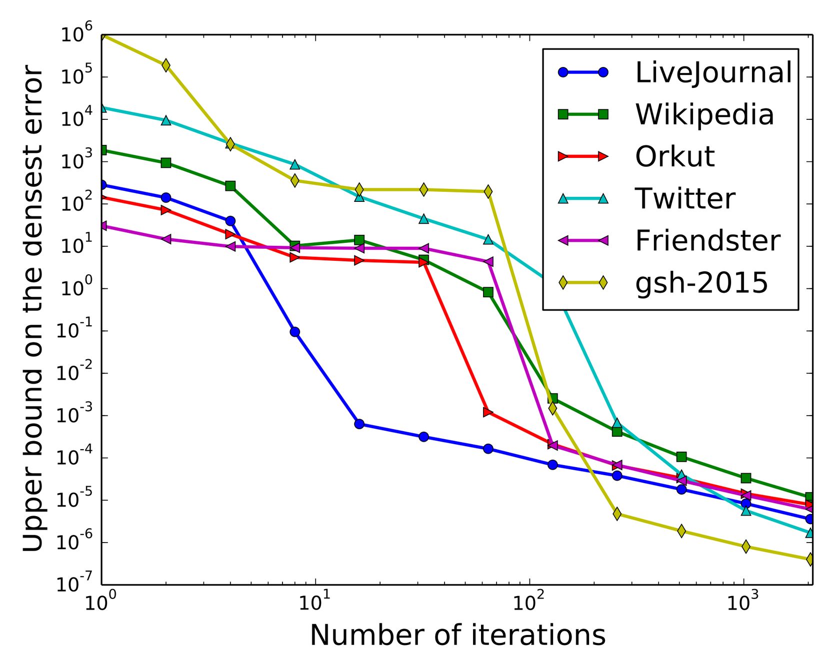
Table 2: Running time comparison of our exact algorithm to the maxflow algorithm of Tatti and Gionis WWW2015.

Networks	exact	TG15
LiveJournal	2m45s	12m02s
Wikipedia	2m14s	7m07s
Orkut	13m08s	1h02m23s
Twitter	4h57m28s	-
Friendster	5h48m27s	-

We use a machine with 64G of RAM for all networks except gsh-2015 for which we use a machine with 512G of RAM.

- Convergence in practice much faster than the worst case one
- 10^{-3} approximation of the densest subgraph within 300 iterations
- 10^{-2} approximation of the full decomposition within 1000 iterations on all networks except gsh-2015 (almost 10^{-1})
- the densest subgraph in gsh-2015 (25G edges) within 10 hours of computation





This research was partially supported by the Hong Kong RGC under the grants 17202715 and 17217716.
This research was partially supported by French National Agency (ANR) under project FIELDS (ANR-15-CE23-0006) and by a Google Faculty Award.