# Fennel: Streaming Graph Partitioning for Massive Scale Graphs

Charalampos E. Tsourakakis [1] Christos Gkantsidis [2]
Bozidar Radunovic [2] Milan Vojnovic [2]

[1] Aalto University, Finland

[2] Microsoft Research, Cambridge UK

MASSIVE 2013, France

Slides available

http://www.math.cmu.edu/~ctsourak/

# Motivation

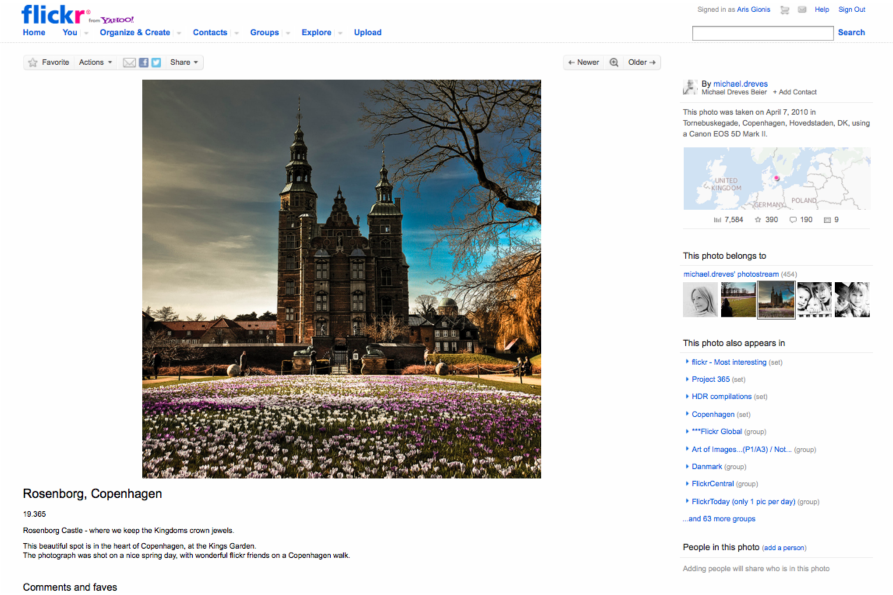
- Big data is data that is too large, complex and dynamic for any conventional data tools to capture, store, manage and analyze.
- The right use of big data allows analysis to spot trends and gives niche insights that help create value and innovation much faster than conventional methods.


Source visual.ly

# Motivation

- We need to handle datasets with billions of vertices and edges
  - `Facebook`: $\sim$ 1 billion users with avg degree 130
  - `Twitter`: $\geq$ 1.5 billion social relations
  - `Google`: web graph more than a trillion edges (2011)
- We need algorithms for dynamic graph datasets
  - real-time story identification using `twitter` posts
  - election trends, `twitter` as election barometer

# Motivation

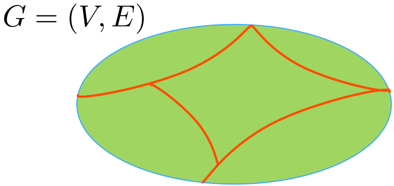

flickr from YAHOO!
Home You Organize & Create Contacts Groups Explore Upload
Signed in as Aris Gionis Help Sign Out
Search
Favorite Actions Share
← Newer Older →
By michael.dreves
Michael Dreves Beier + Add Contact
This photo was taken on April 7, 2010 in Tornebuskegade, Copenhagen, Hovedstaden, DK, using a Canon EOS 5D Mark II.
UNITED KINGDOM
GERMANY
POLAND
7,584 390 190 9
This photo belongs to
michael.dreves' photostream (454)
This photo also appears in
flickr - Most interesting (set)
Project 365 (set)
HDR compilations (set)
Copenhagen (set)
***Flickr Global (group)
Art of Images...(P1/A3) / Not... (group)
Danmark (group)
FlickrCentral (group)
FlickrToday (only 1 pic per day) (group)
...and 63 more groups
People in this photo (add a person)
Adding people will share who is in this photo
Rosenborg, Copenhagen
19.365
Rosenborg Castle - where we keep the Kingdoms crown jewels.
This beautiful spot is in the heart of Copenhagen, at the Kings Garden.
The photograph was shot on a nice spring day, with wonderful flickr friends on a Copenhagen walk.
Comments and faves

# Motivation

- Big graph datasets created from social media data.
  - vertices: photos, tags, users, groups, albums, sets, collections, geo, query, ...
  - edges: upload, belong, tag, create, join, contact, friend, family, comment, fave, search, click, ...
  - also many interesting induced graphs
- What is the underlying graph?
  - tag graph: based on photos
  - tag graph: based on users
  - user graph: based on favorites
  - user graph: based on groups

# Balanced graph partitioning

- Graph has to be distributed across a cluster of machines




- graph partitioning is a way to split the graph vertices in multiple machines
- graph partitioning objectives guarantee low communication overhead among different machines
- additionally balanced partitioning is desirable
- each partition contains $\approx n/k$ vertices, where $n, k$ are the total number of vertices and machines respectively

# Off-line $k$-way graph partitioning

METIS algorithm [Karypis and Kumar, 1998]

- popular family of algorithms and software
- multilevel algorithm
- coarsening phase in which the size of the graph is successively decreased
- followed by bisection (based on spectral or KL method)
- followed by uncoarsening phase in which the bisection is successively refined and projected to larger graphs

METIS is not well understood, i.e., from a theoretical perspective.

# Off-line $k$-way graph partitioning

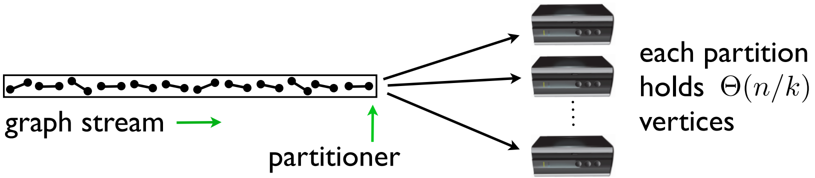
problem: minimize number of edges cut, subject to cluster sizes being at most $\nu n/k$ (bi-criteria approximations)

- $\nu = 2$: Krauthgamer, Naor and Schwartz [Krauthgamer et al., 2009] provide $O(\sqrt{\log k \log n})$ approximation ratio based on the work of Arora-Rao-Vazirani for the sparsest-cut problem ($k = 2$) [Arora et al., 2009]
- $\nu = 1 + \epsilon$: Andreev and Räcke [Andreev and Räcke, 2006] combine recursive partitioning and dynamic programming to obtain $O(\epsilon^{-2} \log^{1.5} n)$ approximation ratio.

There exists a lot of related work, e.g., [Feldmann et al., 2012], [Feige and Krauthgamer, 2002], [Feige et al., 2000] etc.

# streaming *k*-way graph partitioning

- input is a data stream
- graph is ordered
  - arbitrarily
  - breadth-first search
  - depth-first search
- generate an approximately balanced graph partitioning



# Graph representations

- incidence stream
  - at time $t$, a vertex arrives with its neighbors
- adjacency stream
  - at time $t$, an edge arrives

# Partitioning strategies

- hashing: place a new vertex to a cluster/machine chosen uniformly at random
- neighbors heuristic: place a new vertex to the cluster/machine with the maximum number of neighbors
- non-neighbors heuristic: place a new vertex to the cluster/machine with the minimum number of non-neighbors

# Partitioning strategies

[Stanton and Kliot, 2012]

- $d_c(v)$: neighbors of $v$ in cluster $c$
- $t_c(v)$: number of triangles that $v$ participates in cluster $c$

- balanced: vertex v goes to cluster with least number of vertices
- hashing: random assignment
- weighted degree: $v$ goes to cluster $c$ that maximizes $d_c(v) \cdot w(c)$
- weighted triangles: $v$ goes to cluster $j$ that maximizes $t_c(v) / \binom{d_c(v)}{2} \cdot w(c)$

# Weight functions

- $s_c$: number of vertices in cluster $c$
- unweighted: $w(c) = 1$
- linearly weighted: $w(c) = 1 - s_c(k/n)$
- exponentially weighted: $w(c) = 1 - e^{(s_c - n/k)}$

# FENNEL algorithm

The standard formulation hits the ARV barrier

$$\begin{aligned} \text{minimize}_{\mathcal{P}=(S_1,\dots,S_k)} \quad & |\partial\, e(\mathcal{P})| \\ \text{subject to} \quad & |S_i| \le \nu \frac{n}{k}, \text{ for all } 1 \le i \le k \end{aligned}$$

- We relax the hard cardinality constraints

$$\text{minimize}_{\mathcal{P}=(S_1,\dots,S_k)} \quad |\partial\, E(\mathcal{P})| + c_{\mathrm{IN}}(\mathcal{P})$$

where $c_{\mathrm{IN}}(\mathcal{P}) = \sum_i s(|S_i|)$, so that objective self-balances

# FENNEL algorithm

- for $S \subseteq V$, $f(S) = e[S] - \alpha|S|^\gamma$, with $\gamma \geq 1$
- given partition $\mathcal{P} = (S_1, \ldots, S_k)$ of $V$ in $k$ parts define

$$g(\mathcal{P}) = f(S_1) + \ldots + f(S_k)$$

- the goal: maximize $g(\mathcal{P})$ over all possible $k$-partitions
- notice:

$$g(\mathcal{P}) = \underbrace{\sum_i e[S_i]}_{\substack{m-\text{number of}\\ \text{edges cut}}} - \underbrace{\alpha \sum_i |S_i|^\gamma}_{\substack{\text{minimized for}\\ \text{balanced partition!}}}$$

# Connection

notice

$$f(S) = e[S] - \alpha \binom{|S|}{2}$$

- related to modularity
- related to optimal quasicliques [Tsourakakis et al., 2013]

# FENNEL algorithm

## Theorem

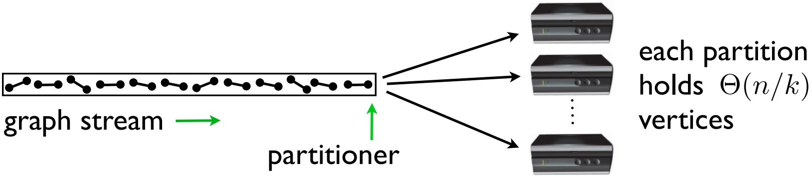
- For $\gamma = 2$ there exists an algorithm that achieves an approximation factor $\log(k)/k$
  for a shifted objective where $k$ is the number of clusters
  - semidefinite programming algorithm
  - in the shifted objective the main term takes care of the load balancing and the second order term minimizes the number of edges cut
  - Multiplicative guarantees not the most appropriate
- random partitioning gives approximation factor $1/k$
- no dependence on $n$

  mainly because of relaxing the hard cardinality constraints

# FENNEL algorithm — greedy scheme

- $\gamma = 2$ gives non-neighbors heuristic
- $\gamma = 1$ gives neighbors heuristic
- interpolate between the two heuristics, e.g., $\gamma = 1.5$

# FENNEL algorithm — greedy scheme



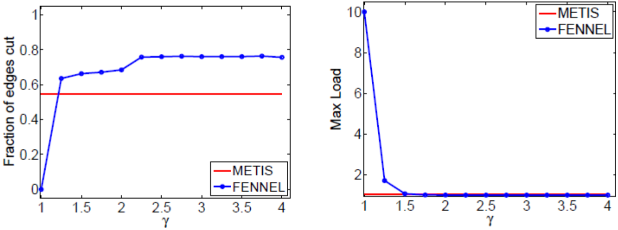

- send $v$ to the partition / machine that maximizes

$$
\begin{aligned}
f(S_i \cup \{v\}) - f(S_i) \\
&= e[S_i \cup \{v\}] - \alpha(|S_i| + 1)^\gamma - (e[S_i] - \alpha|S_i|^\gamma) \\
&= d_{S_i}(v) - \alpha\mathcal{O}(|S_i|^{\gamma-1})
\end{aligned}
$$

- fast, amenable to streaming and distributed setting

# FENNEL algorithm — $\gamma$

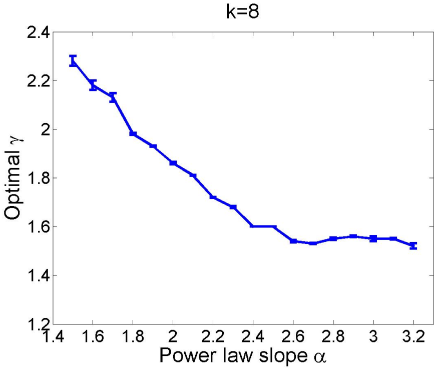
Explore the tradeoff between the number of edges cut and load balancing.




Fraction of edges cut $\lambda$ and maximum load normalized $\rho$ as a function of $\gamma$, ranging from 1 to 4 with a step of 0.25, over five randomly generated power law graphs with slope 2.5. The straight lines show the performance of METIS.

- Not the end of the story ... choose $\gamma^*$ based on some "easy-to-compute" graph characteristic.

# FENNEL algorithm — $\gamma^*$




y-axis Average optimal value $\gamma^*$ for each power law slope in the range $[1.5, 3.2]$ using a step of 0.1 over twenty randomly generated power law graphs that results in the smallest possible fraction of edges cut $\lambda$ conditioning on a maximum normalized load $\rho = 1.2$, $k = 8$. x-axis Power-law exponent of the degree sequence. Error bars indicate the variance around the average optimal value $\gamma^*$.

# FENNEL algorithm — results

Twitter graph with approximately 1.5 billion edges, $\gamma = 1.5$

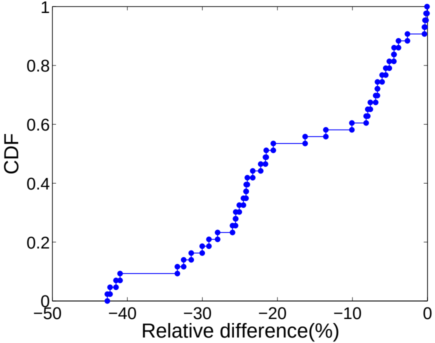
$$\lambda = \frac{\#\{\text{edges cut}\}}{m} \qquad \rho = \max_{1 \le i \le k} \frac{|S_i|}{n/k}$$

| | Fennel | | Best competitor | | Hash Partition | | METIS | |
|---|---|---|---|---|---|---|---|---|
| $k$ | $\lambda$ | $\rho$ | $\lambda$ | $\rho$ | $\lambda$ | $\rho$ | $\lambda$ | $\rho$ |
| 2 | 6.8% | 1.1 | 34.3% | 1.04 | 50% | 1 | 11.98% | 1.02 |
| 4 | 29% | 1.1 | 55.0% | 1.07 | 75% | 1 | 24.39% | 1.03 |
| 8 | 48% | 1.1 | 66.4% | 1.10 | 87.5% | 1 | 35.96% | 1.03 |

Table: Fraction of edges cut $\lambda$ and the normalized maximum load $\rho$ for Fennel, the best competitor and hash partitioning of vertices for the Twitter graph. Fennel and best competitor require around 40 minutes, METIS more than $8\frac{1}{2}$ hours.

# FENNEL algorithm — results

Extensive experimental evaluation over $>40$ large real graphs [Tsourakakis et al., 2012]




CDF of the relative difference $\frac{\lambda_{fennel} - \lambda_c}{\lambda_c} \times 100\%$ of percentages of edges cut of FENNEL and the best competitor (pointwise) for all graphs in our dataset.

# FENNEL algorithm — "zooming in"

Performance of various existing methods on amazon0312 for $k = 32$

| | BFS | | Random | |
|---|---|---|---|---|
| Method | $\lambda$ | $\rho$ | $\lambda$ | $\rho$ |
| H | 96.9% | 1.01 | 96.9% | 1.01 |
| B [Stanton and Kliot, 2012] | 97.3% | 1.00 | 96.8% | 1.00 |
| DG [Stanton and Kliot, 2012] | 0% | 32 | 43% | 1.48 |
| LDG [Stanton and Kliot, 2012] | 34% | 1.01 | 40% | 1.00 |
| EDG [Stanton and Kliot, 2012] | 39% | 1.04 | 48% | 1.01 |
| T [Stanton and Kliot, 2012] | 61% | 2.11 | 78% | 1.01 |
| LT [Stanton and Kliot, 2012] | 63% | 1.23 | 78% | 1.10 |
| ET [Stanton and Kliot, 2012] | 64% | 1.05 | 79% | 1.01 |
| NN [Prabhakaran and et al., 2012] | 69% | 1.00 | 55% | 1.03 |
| Fennel | 14% | 1.10 | 14% | 1.02 |
| METIS | 8% | 1.00 | 8% | 1.02 |

# Conclusions

summary and future directions

- cheap and efficient graph partitioning is highly desired
- new area [Stanton and Kliot, 2012], [Tsourakakis et al., 2012], [Nishimura and Ugander, 2013]
- average case analysis
- stratified graph partitioning [Nishimura and Ugander, 2013]

# thank you!

# references I


Andreev, K. and Räcke, H. (2006).
Balanced graph partitioning.
*Theor. Comp. Sys.*, 39(6):929–939.

Arora, S., Rao, S., and Vazirani, U. (2009).
Expander flows, geometric embeddings and graph partitioning.
*Journal of the ACM (JACM)*, 56(2).

Feige, U. and Krauthgamer, R. (2002).
A polylogarithmic approximation of the minimum bisection.
*SIAM Journal on Computing*, 31(4):1090–1118.

Feige, U., Krauthgamer, R., and Nissim, K. (2000).
Approximating the minimum bisection size.
In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 530–536. ACM.

# references II


Feldmann, A. E., Foschini, L., et al. (2012).
Balanced partitions of trees and applications.
In *Symposium on Theoretical Aspects of Computer Science*, volume 14, pages 100–111.

Karypis, G. and Kumar, V. (1998).
A fast and high quality multilevel scheme for partitioning irregular graphs.
*SIAM J. Sci. Comput.*, 20(1):359–392.

Krauthgamer, R., Naor, J. S., and Schwartz, R. (2009).
Partitioning graphs into balanced components.
In *SODA*.

# references III

Nishimura, J. and Ugander, J. (2013).

Restreaming graph partitioning: simple versatile algorithms for advanced balancing.

In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1106–1114. ACM.

Prabhakaran, V. and et al. (2012).

Managing large graphs on multi-cores with graph awareness.

In *USENIX ATC'12*.

Stanton, I. and Kliot, G. (2012).

Streaming graph partitioning for large distributed graphs.

In *KDD*.

# references IV


Tsourakakis, C. E., Bonchi, F., Gionis, A., Gullo, F., and Tsiarli, M. A. (2013).
Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees.
*KDD*.

Tsourakakis, C. E., Gkantsidis, C., Radunovic, B., and Vojnovic, M. (2012).
FENNEL: Streaming graph partitioning for massive scale graphs.
Technical report.