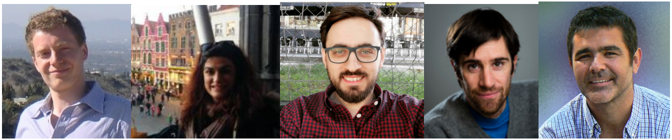
Perturb and Combine to Identify Influential Spreaders in Real-World Networks

Antoine Tixier[1], Maria Rossi[1], Fragkiskos Malliaros[2], Jesse Read[1], Michalis Vazirgiannis[1]
[1]**DaSciM team**, École Polytechnique; [2]CentraleSupélec



Oral at ASONAM'19, Vancouver, Canada, Aug 28, 2019

Contact: `antoine.tixier-1@colorado.edu`

Paper: https://arxiv.org/pdf/1807.09586.pdf

Agenda

1. Motivation
2. P&C for networks
3. Social networks
4. Word co-occurrence networks
5. Conclusion

Influential spreader detection

Influential spreaders: *nodes that can diffuse information to the largest part of the network in a given amount of time.*
Influential spreader detection can be broken down into:

- ✓ identifying **individual** influential nodes
- ✓ influence maximization: identifying a **group** of nodes that together maximize the total spread of influence

→ here, we focus on the identification of **individual** influential nodes

Many important applications: epidemiology, viral marketing, social media analysis, expert finding, NLP...

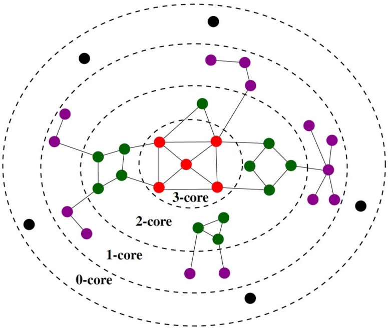
Graph degeneracy (Seidman 1983)

- △ **k-core of $G(V, E)$**: maximal subgraph of G in which every vertex v has at least degree k
- △ **core number of $v \in V$**: highest order of a k-core that contains v
- △ **very fast**: $\mathcal{O}(|V| + |E|)$ and $\mathcal{O}(|E| \log(|V|))$ in weighted case (Batagelj and Zaveršnik 2002)

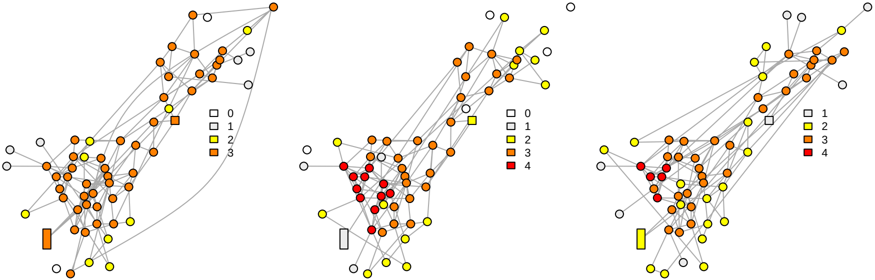
Key facts:

+ core numbers **correlate well with spreading influence**, and much better than degrees or PageRank scores (Kitsak et al. 2010)

− k-cores are **unstable** to perturbations (Adiga and Vullikanti 2013; Goltsev, Dorogovtsev, and Mendes 2006).



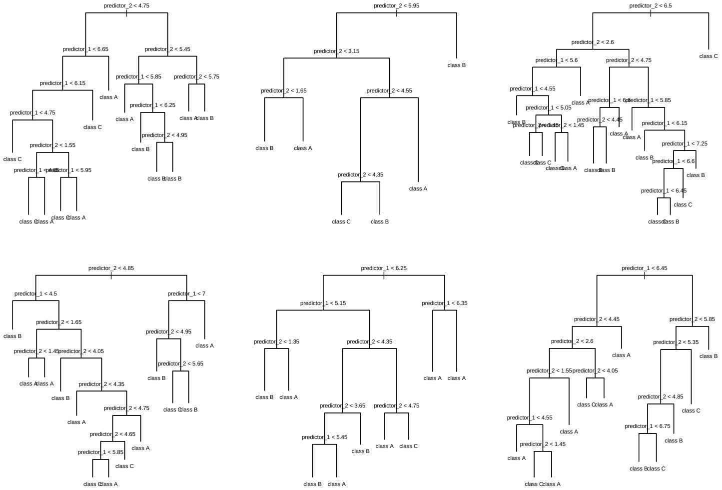
Instability of graph degeneracy



node	original	pert. #1	pert. #2	pert. #3
square	4	3	2	1
rectangle	1	3	1	2

Link with unstable learners

Decision trees are unstable to small perturbations of their training set:



Perturb and combine in machine learning (1/2)

- △ **unstable learners**: small changes in training set → large changes in predictions
- △ a.k.a. **strong learners** or **low bias-high variance** algorithms (Breiman 1996b)
- △ e.g., unpruned decision trees

Key fact: well known that **Perturb and Combine** (P&C) strategies boost the performance of unstable learners

Most famous example: **b**ootstrap **agg**regat**ing** (**bagging**) (Breiman 1996a), at the core of Random Forest (Breiman 2001)

Perturb and Combine in machine learning (2/2)

Most famous P&C approach: bagging

- △ bootstrap samples are generated by **perturbing** the training set (drawing with replacement)
- △ unpruned trees are trained **in parallel** on the bootstrap samples
- △ individual predictions are **combined** through averaging or voting

P&C works mainly by **reducing the variance** of high variance-low bias algorithms (Breiman 1996a). It cannot help with low-variance algorithms, e.g., k nearest-neighbors.

Our idea: perturb and combine for networks

Recap

- graph degeneracy is very effective at locating influential spreaders, but unstable
- in ML, P&C is known to boost unstable models

$\rightarrow$ Our objective is to show that:

"Like unstable learners, degeneracy-based node scoring functions, and more generally any unstable node scoring function, benefits from P&C"

More precisely:

"One can identify better spreaders by aggregating node scores computed on multiple perturbed versions of the original network rather than by using the scores computed on the original network"

Perturb and combine for networks

P&C for networks

- **perturb**: create n perturbed versions of the original network
- **mine**: apply a node scoring function to each perturbed network,
- **combine**: combine the results.

P&C for networks is **trivially parallelizable**
$\rightarrow$ P&C scores do not take more time to obtain than the original scores, provided that n workers are available

Perturb step

Edge-based perturbation scheme (Adiga and Vullikanti 2013)

Let $G(V, E)$ be the original graph and $\mathbb{G}$ be a random graph model.

- △ if edge (u, v) already exists, it is deleted with some probability
- △ if it does not exist, it is added with some probability
- △ variant in which edge weights are incremented/decremented

probabilities are given by $\mathbb{G}$

Random graph models

- **uniform perturbation** with the Erdős-Rényi (ER) model (Erdös and Rényi 1960)
- **degree assortative perturbation** with the Chung-Lu (CL) model (Chung and Lu 2002)

Mine and combine steps

Mine

- ✓ since P&C in machine learning is most effective when used with *unstable* learners, we experimented with ***k*-core** and **weighted *k*-core**
- ✓ we also tried with **PageRank** (Page et al. 1999), supposedly more stable (Ipsen and Wills n.d.; Ng, Zheng, and Jordan 2001)

Combine

We use **averaging**, like in bagging regression trees

Social networks: experiments

	$\|V\|$	$\|E\|$	*diameter*
EMAIL-ENRON	33,696	180,811	11
EPINIONS	75,877	405,739	14
WIKI-VOTE	7,066	100,736	7

Experimental setup (F. D. Malliaros, Rossi, and Vazirgiannis 2016)

- we compare the **average severity** of the epidemic when started from the top nodes in terms of original scores/P&C scores
- epidemics are simulated with **SIR** (Kermack and McKendrick 1932)
- results are averaged over N_e **epidemics** started from each node in the trigger population* and over all nodes in that population

*main core (for unweighted and weighted k-core) or top 100 nodes (for PageRank)

we assigned as edge weights the max degree of their endpoints

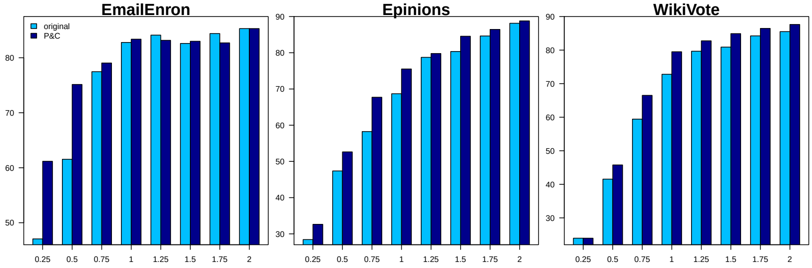
Social networks: results (1/2)

	Network	Scores	2	4	6	8	10	Total	+%
			Time Step						
Unweighted *k*-c	Enron	**P&C**	**16**	**89**	**300**	**419**	**269**	**2,538**	3.76
		original	14	77	269	401	275	2,446	
	Epinions	**P&C**	**8**	**34**	**110**	**245**	**317**	**2,436**	4.35
		original	7	30	100	224	301	2,330	
	WikiVote	**P&C**	**3**	**8**	**17**	**29**	**40**	**490**	3.47
		original	3	8	16	28	37	473	
Weighted *k*-c	Enron	**P&C**	**26**	**141**	**407**	**445**	**226**	**2,724**	3.52
		original	20	110	345	433	253	2,628	
	Epinions	**P&C**	**11**	**46**	**146**	**302**	**353**	**2,689**	2.42
		original	11	42	135	286	345	2,624	
	WikiVote	**P&C**	**5**	**12**	**24**	**39**	**50**	**612**	19.3
		original	4	9	18	31	42	513	
PageRank	Enron	**P&C**	**16**	**86**	**278**	**389**	**266**	**2,454**	4.93
		original	15	80	259	366	255	2,333	
	Epinions	**P&C**	**11**	**42**	**132**	**276**	**336**	**2,598**	2.04
		original	11	41	127	267	326	2,545	
	WikiVote	**P&C**	**5**	**11**	**22**	**38**	**49**	**596**	2.35
		original	5	11	22	36	48	582	

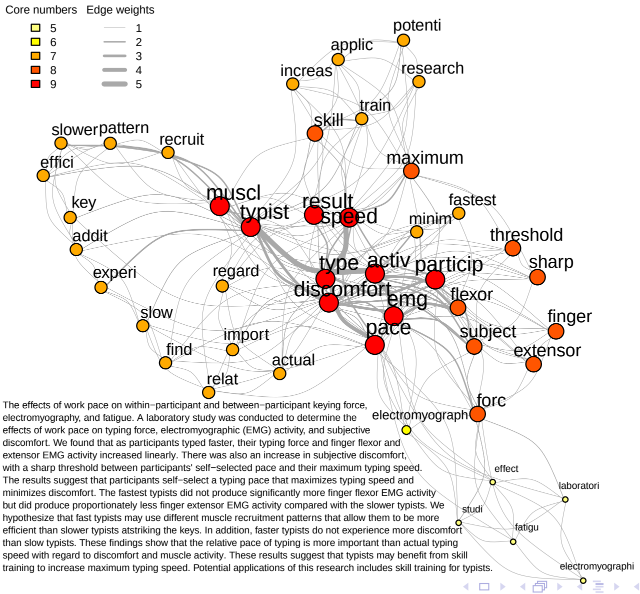
Social networks: results (2/2)

Ranking comparison for unweighted k-core

How much of the p% best spreaders in terms of SIR are present in the top p% nodes in terms of original and P&C scores?



Word co-occurrence networks (Mihalcea and Tarau 2004)



Word co-occurrence networks: experiments

keywords are **influential nodes** within the word co-occurrence network of their document (Tixier, F. Malliaros, and Vazirgiannis 2016).
Does P&C on graphs of words improve keyword extraction?

Experimental setup

- Hulth 2003 dataset of 500 research paper abstracts:
  - $\sim$ 120 words/document
  - $\sim$ 21 keywords from human annotators/document on average
  - $\sim$ # of nodes, edges, and diameter: 32, 155, and 3.6
- for unweighted and weighted k-core, keywords as **main core**
- for PageRank, keywords as **top 33% nodes**

Word co-occurrence networks: results

		scores	precision	recall	F1-score	+%
	unweighted *k*-core	**P&C**	**52.09**	**51.25**	**54.88**	5.70
		original	48.76	46.90	51.75	
	weighted *k*-core	**P&C**	**50.53**	**48.54**	**52.50**	7.45
		original	48.07	46.81	48.86	
	PageRank	**P&C**	**45.53**	**42.73**	**46.75**	2.33
		original	45.21	41.89	45.66	
SOTA		[Tixier16]	48.79	72.78	56.00	
SOTA		[Rousseau15]	61.24	50.32	51.92	
SOTA		[Mihalcea04]	51.95	54.99	50.40	

Conclusion

Contributions

- we proposed one of the **first applications** of P&C to networks
- we showed that P&C scores identify **better spreaders** than the original scores
- P&C for networks is **trivially parallelizable**
- our framework is general and can be used with **other graph mining algorithms** and applied to other tasks (e.g., **community detection**)

What's more in the paper?

- **theoretical analysis**:
  - define bias and variance of node scoring function
  - demonstrate that P&C reduces error
- more details and experiments

Thank you for your attention!

Questions? → `antoine.tixier-1@colorado.edu`
Paper: https://arxiv.org/pdf/1807.09586.pdf

References I

Adiga, Abhijin and Anil Kumar S Vullikanti (2013). "How robust is the core of a network?". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 541–556.

Batagelj, Vladimir and Matjaž Zaveršnik (2002). "Generalized cores". In: *arXiv preprint cs/0202039*.

Breiman, Leo (1996a). "Bagging predictors". In: *Machine learning* 24.2, pp. 123–140.

– (1996b). "Bias, variance, and arcing classifiers". In:

– (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Chung, Fan and Linyuan Lu (2002). "The average distances in random graphs with given expected degrees". In: *Proceedings of the National Academy of Sciences* 99.25, pp. 15879–15882.

Erdös, P. and A. Rényi (1960). "On the evolution of random graphs". In: *Publ. Math. Inst. Hung. Acad. Sci* 5, pp. 17–61.

References II

- Goltsev, Alexander V, Sergey N Dorogovtsev, and Jose Ferreira F Mendes (2006). "k-core (bootstrap) percolation on complex networks: Critical phenomena and nonlocal effects". In: *Physical Review E* 73.5, p. 056101.
- Hulth, Anette (2003). "Improved automatic keyword extraction given more linguistic knowledge". In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp. 216–223.
- Ipsen, Ilse CF and Rebecca S Wills (n.d.). "Mathematical properties and analysis of Google's PageRank". In:
- Kermack, William O and Anderson G McKendrick (1932). "Contributions to the mathematical theory of epidemics. II. The problem of endemicity". In: vol. 138. 834. JSTOR, pp. 55–83.
- Kitsak, Maksim et al. (2010). "Identification of influential spreaders in complex networks". In: *Nature physics* 6.11, pp. 888–893.

References III

- Malliaros, Fragkiskos D, Maria-Evgenia G Rossi, and Michalis Vazirgiannis (2016). "Locating influential nodes in complex networks". In: *Scientific reports* 6, p. 19307.
- Mihalcea, Rada and Paul Tarau (2004). "TextRank: bringing order into texts". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Ng, Andrew Y, Alice X Zheng, and Michael I Jordan (2001). "Link analysis, eigenvectors and stability". In: *International Joint Conference on Artificial Intelligence*. Vol. 17. 1. LAWRENCE ERLBAUM ASSOCIATES LTD, pp. 903–910.
- Page, Lawrence et al. (1999). *The PageRank citation ranking: Bringing order to the web.*. Tech. rep. Stanford InfoLab.
- Rousseau, François and Michalis Vazirgiannis (2015). "Main Core Retention on Graph-of-Words for Single-Document Keyword Extraction". In: *Advances in Information Retrieval*. Springer, pp. 382–393.

References IV

Seidman, Stephen B (1983). "Network structure and minimum degree". In: *Social networks* 5.3, pp. 269–287.

Tixier, Antoine, Fragkiskos Malliaros, and Michalis Vazirgiannis (2016). "A graph degeneracy-based approach to keyword extraction". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1860–1870.