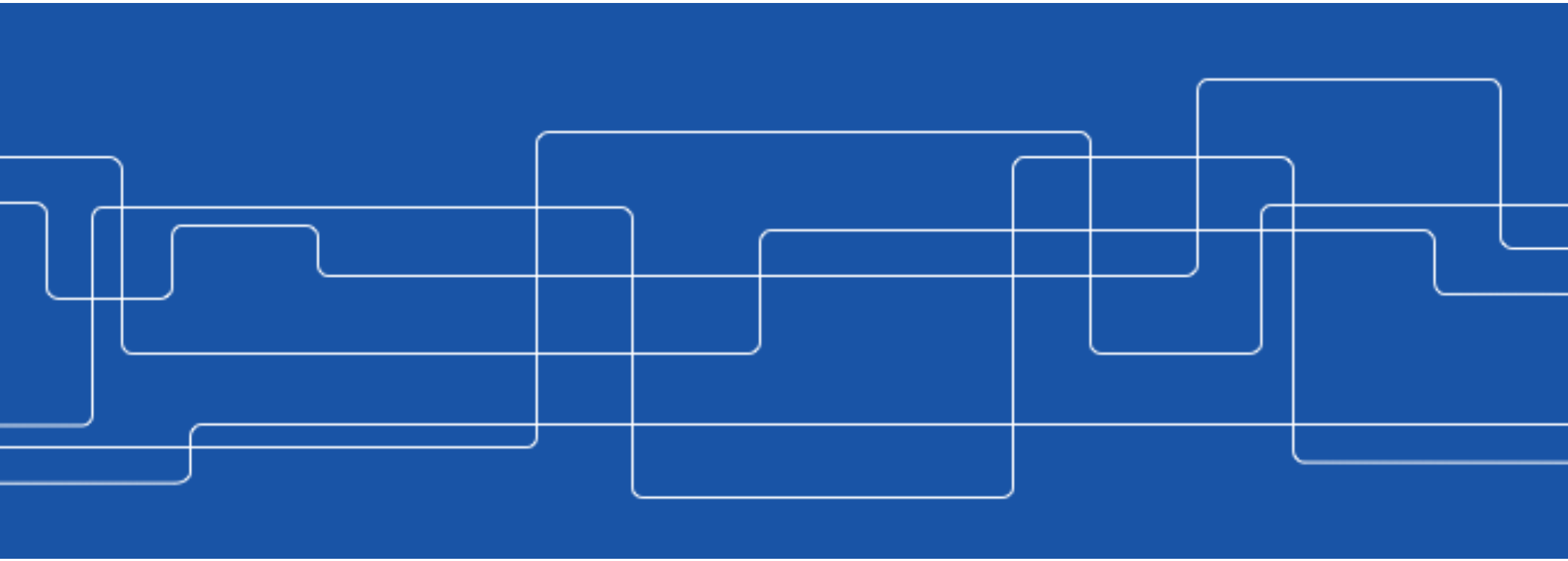
Node-Cut Partitioning of Dense, Weighted Graphs with Application to Topic Detection in Text

Kambiz Ghoorchian
Šarūnas Girdzijauskas

ghoorian@kth.se
28.01.2016



Topic Detection in Text

Given a large number of **documents** (e.g., Tweets), how can we extract the most **frequent** (significant) **topics**.

Topic Detection in Text

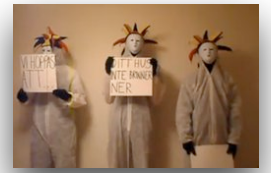
Given a large number of **documents** (e.g., Tweets), how can we extract the most **frequent** (significant) **topics**.

- Sample topics extracted from a Tweeter dataset, collected in geographical area of "Stockholm", during the Swedish presidential election in 2014:
  - Job Market, Young People

  - Foreign policy - Immigration - Russia - Ukraine

  - Swedish reporter, “Anna Hedenmo”, laughed at “Stephan Löfven” (presidential Candidate).

  - A threatening DVD-Film sent to “Urban Ahlin”, a Swedish social democratic politician.



Topic Detection in Text

- Existing works

    1. Vector based modeling - Pairwise Similarity comparison
    2. Statistical Topic modeling (Dimensionality Reduction)
        1. Latent Semantic Analysis (LSA)
        2. Singular Value Decomposition (SVD)
        3. Non-negative Matrix Factorization

Topic Detection in Text

- Existing works

1. Vector based modeling - Pairwise Similarity comparison
2. Statistical Topic modeling (Dimensionality Reduction)
    1. Latent Semantic Analysis (LSA)
    2. Singular Value Decomposition (SVD)
    3. Non-negative Matrix Factorization

Scalability ?

Topic Detection in Text

- Existing works

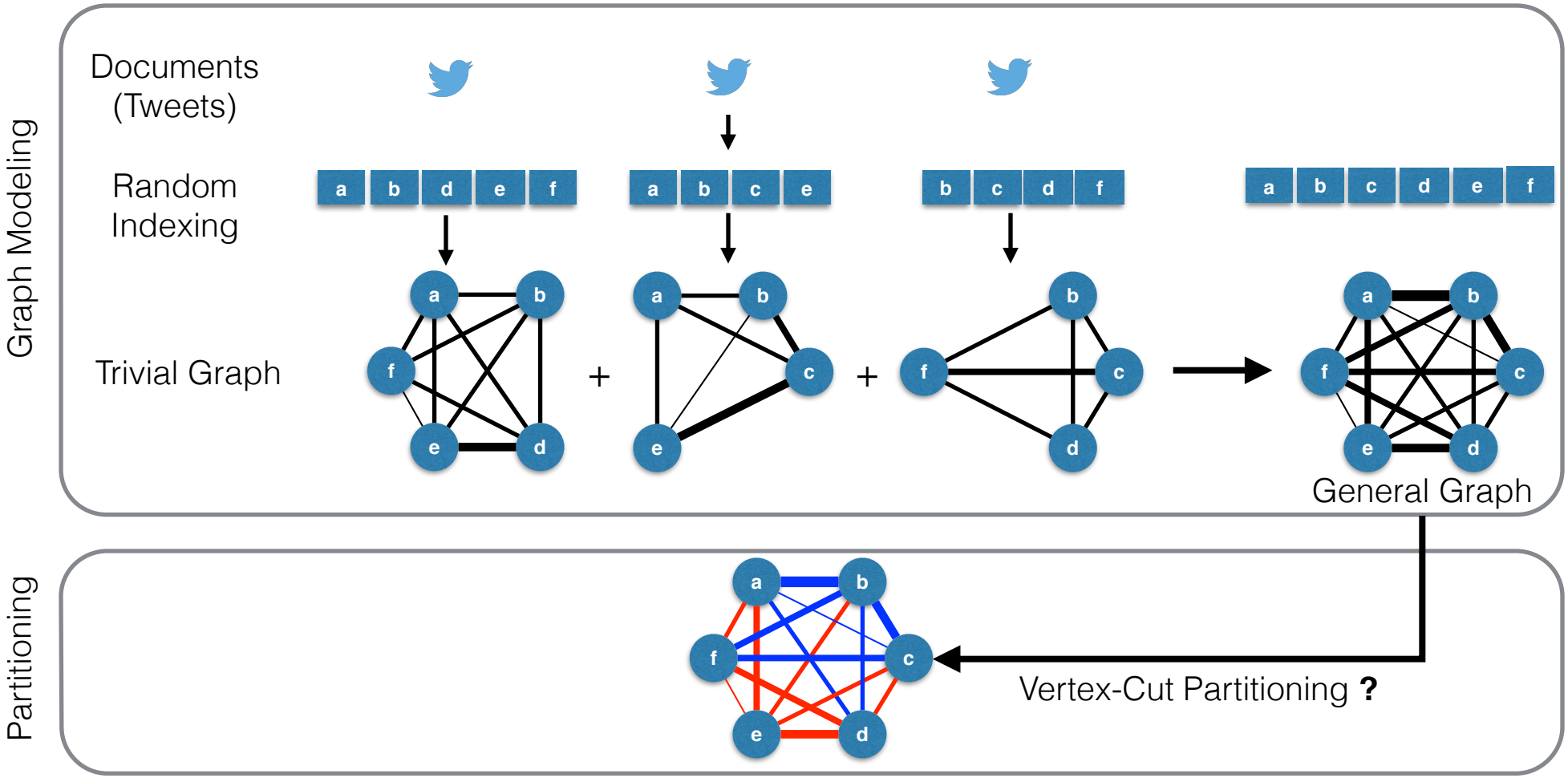
1. Vector based modeling - Pairwise Similarity comparison
2. Statistical Topic modeling (Dimensionality Reduction)
    1. Latent Semantic Analysis (LSA)
    2. Singular Value Decomposition (SVD)
    3. Non-negative Matrix Factorization
3. Graph based Modeling and Community Detection[1]

Scalability ?

1. K Ghoorchian, F Rahimian, S Girdzijauskas: Semi Supervised Multiple Disambiguation, Trustcom/BigDataSE/ISPA, 2015 IEEE 2, 88-95.

The protocol

Based on **Distributional Semantics** and **Graph analytics**.



Random Indexing

- Dimensionality Reduction Method (SVD, Hashing)
  - Inspired from Sparse Distributed Memory (SDM) [1]
  - Random Projection Models [2]
- Based on **Johansson & Lindenstrauss's (JL) lemma** [3]

  "For any set of ***P*** vectors in a high **n-dimensional** euclidian space there exists a mapping onto an ***m-dimensional*** space

$$m \geq m_0 = \mathrm{O}(^{\log p}/_{\epsilon^2})$$

  that does not distort the distances between any pair of vectors with high probability, by a factor more than

$$1 \pm \epsilon.$$

1. Kanevara, P: Sparse Distributed Memory and Related Models. Associative Neural Memories, Oxford University Press, 1993.
2. Kanerava, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In Gleitman, L. R. and Josh, A. K., editors, *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, page 1036, Mahwah, New Jersey. Erlbaum.
3. Johnson, W. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hil- bert space. In Beals, R., Beck, A., Bellow, A., and Hajian, A., editors, *Conference on Modern Analysis and Probability (1982: Yale University)*, volume 26 of *Con- temporary Mathematics*, pages 189–206. American Mathematical Society.

Random Indexing

- Example

Documents

D1 = {W11, W12, W13,…}

D2 = {W21, W22, W23,…}

D3 = {W31, W32, W33,…}

…

Dn = {Wn1, Wn2, Wn3,…}

Random Indexing

- Example

Documents

D1 = {W11, W12, W13,…}

D2 = {W21, W22, W23,…}

D3 = {W31, W32, W33,…}

…

Dn = {Wn1, Wn2, Wn3,…}

Word Space Modeling

Term - Doc - Matrix

	w11	w12	w13	w21	w22	w23	w31	w32	w33	…	Wnm
D1	1	1	1	0	0	0	0	0	0	0	0
D2	0	0	0	1	1	1	0	0	0	0	0
D3	0	0	0	0	0	0	1	1	1	0	0
…											
Dn	0	0	0	0	0	0	0	0	0	1	1

m

Random Indexing

- Example

Documents

D1 = {W11, W12, W13,…}

D2 = {W21, W22, W23,…}

D3 = {W31, W32, W33,…}

…

Dn = {Wn1, Wn2, Wn3,…}

Word Space Modeling

Term - Doc - Matrix

	w11	w12	w13	w21	w22	w23	w31	w32	w33	…	Wnm
D1	1	1	1	0	0	0	0	0	0	0	0
D2	0	0	0	1	1	1	0	0	0	0	0
D3	0	0	0	0	0	0	1	1	1	0	0
…											
Dn	0	0	0	0	0	0	0	0	0	1	1

m

Random Indexing

W_{11} →

V11	1	0	1	1	1	0	0	0	1	0

d

Random Indexing

- Example

Documents

D1 = {W11, W12, W13,…}

D2 = {W21, W22, W23,…}

D3 = {W31, W32, W33,…}

…

Dn = {Wn1, Wn2, Wn3,…}

Word Space Modeling

Term - Doc - Matrix

	w11	w12	w13	w21	w22	w23	w31	w32	w33	…	Wnm
D1	1	1	1	0	0	0	0	0	0	0	0
D2	0	0	0	1	1	1	0	0	0	0	0
D3	0	0	0	0	0	0	1	1	1	0	0
…											
Dn	0	0	0	0	0	0	0	0	0	1	1

m

Random Indexing

W_{11} →

Word Vectors

V11	1	0	1	1	1	0	0	0	1	0
V12	0	1	1	0	1	0	0	1	1	0
V13	1	0	0	0	1	1	1	1	0	0
V21	0	1	0	0	1	0	1	0	1	1
…										
Vm	0	0	0	1	1	1	1	0	0	1

d

Random Indexing

- Example

Documents

D1 = {W11, W12, W13,…}

D2 = {W21, W22, W23,…}

D3 = {W31, W32, W33,…}

…

Dn = {Wn1, Wn2, Wn3,…}

Word Space Modeling

Term - Doc - Matrix

	w11	w12	w13	w21	w22	w23	w31	w32	w33	…	Wnm
D1	1	1	1	0	0	0	0	0	0	0	0
D2	0	0	0	1	1	1	0	0	0	0	0
D3	0	0	0	0	0	0	1	1	1	0	0
						…					
Dn	0	0	0	0	0	0	0	0	0	1	1

m

Random Indexing

W_{11} →

Word Vectors

V11	1	0	1	1	1	0	0	0	1	0
V12	0	1	1	0	1	0	0	1	1	0
V13	1	0	0	0	1	1	1	1	0	0
V21	0	1	0	0	1	0	1	0	1	1
…										
Vm	0	0	0	1	1	1	1	0	0	1

D1	2	2	2	1	3	1	1	2	2	0

d

Random Indexing

- Example

Documents

D1 = {W11, W12, W13,…}

D2 = {W21, W22, W23,…}

D3 = {W31, W32, W33,…}

…

Dn = {Wn1, Wn2, Wn3,…}

Word Space Modeling

Term - Doc - Matrix

	w11	w12	w13	w21	w22	w23	w31	w32	w33	…	Wnm
D1	1	1	1	0	0	0	0	0	0	0	0
D2	0	0	0	1	1	1	0	0	0	0	0
D3	0	0	0	0	0	0	1	1	1	0	0
					…						
Dn	0	0	0	0	0	0	0	0	0	1	1

m

Random Indexing

W_{11} →

Word Vectors

V11	1	0	1	1	1	0	0	0	1	0
V12	0	1	1	0	1	0	0	1	1	0
V13	1	0	0	0	1	1	1	1	0	0
V21	0	1	0	0	1	0	1	0	1	1
…										
Vm	0	0	0	1	1	1	1	0	0	1

Doc Vectors

D1	2	2	2	1	3	1	1	2	2	0
D2	0	5	5	0	0	0	0	5	7	0
D3	0	0	0	0	3	3	1	0	0	0
					…					
Dn	2	1	2	1	2	0	1	3	6	0

d

Random Indexing

- Example

Documents

D1 = {W11, W12, W13,…}

D2 = {W21, W22, W23,…}

D3 = {W31, W32, W33,…}

…

Dn = {Wn1, Wn2, Wn3,…}

Word Space Modeling

Term - Doc - Matrix

	w11	w12	w13	w21	w22	w23	w31	w32	w33	…	Wnm
D1	1	1	1	0	0	0	0	0	0	0	0
D2	0	0	0	1	1	1	0	0	0	0	0
D3	0	0	0	0	0	0	1	1	1	0	0
					…						
Dn	0	0	0	0	0	0	0	0	0	1	1

m

Random Indexing

W_{11} →

Word Vectors

V11	1	0	1	1	1	0	0	0	1	0
V12	0	1	1	0	1	0	0	1	1	0
V13	1	0	0	0	1	1	1	1	0	0
V21	0	1	0	0	1	0	1	0	1	1
…										
Vm	0	0	0	1	1	1	1	0	0	1

Doc Vectors

D1	2	2	2	1	3	1	1	2	2	0
D2	0	5	5	0	0	0	0	5	7	0
D3	0	0	0	0	3	3	1	0	0	0
					…					
Dn	2	1	2	1	2	0	1	3	6	0

d

Advantages

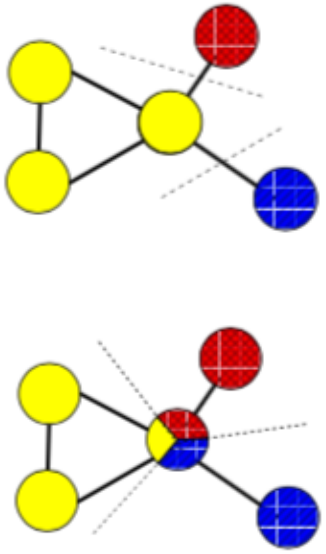
1 - RI is Incremental

2 - d is a parameter

3 - d << m

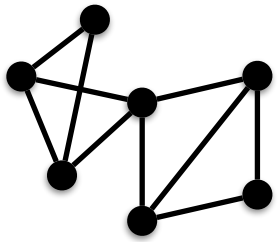
Vertex-Cut Partitioning

- Ja-Be-Ja-VC[1]
  - Balanced k-way Partitioning
  - Un-weighted
  - Iterative
  - Local Search & Optimization
  - Node Cut-Minimization
  - Simulated Annealing
  - Parameters
    - K number of partitions
    - H heat factor



1. F Rahimian, AH Payberah, S Girdzijauskas, S Haridi: Distributed Vertex-cut Partitioning, in Distributed Applications and Interoperable Systems, 186-200, 2014.

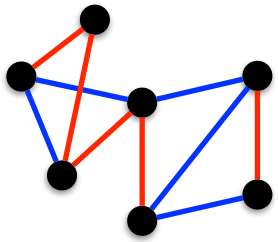
Vertex-Cut Partitioning



Random Initialization

Vertex-Cut Partitioning

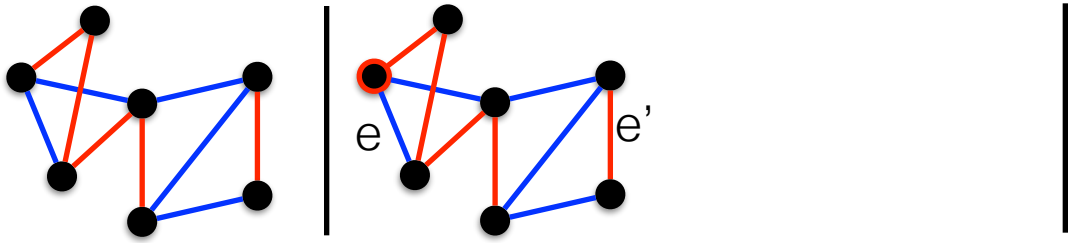
k = 2



Random Initialization

Vertex-Cut Partitioning

k = 2



Random Initialization

Iteration

C = Blue
C' = Red

$$utility = ((v(e,c') + v(e',c)) \times T_r) - (v(e,c) + v(e',c'))$$

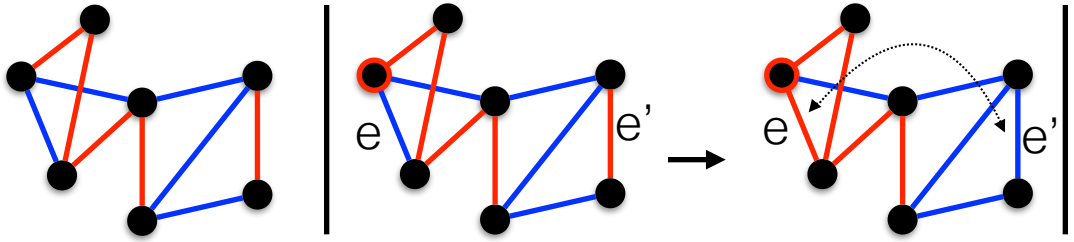
Gain

Heat

$$v(e,c') = \frac{\sum_{C_i \in N_e, C_{i=}c'} C_i}{|C_{N_e}|}$$

Vertex-Cut Partitioning

k = 2



Random Initialization

Iteration

C = Blue
C' = Red

$$utility = ((v(e,c') + v(e',c)) \times T_r) - (v(e,c) + v(e',c'))$$

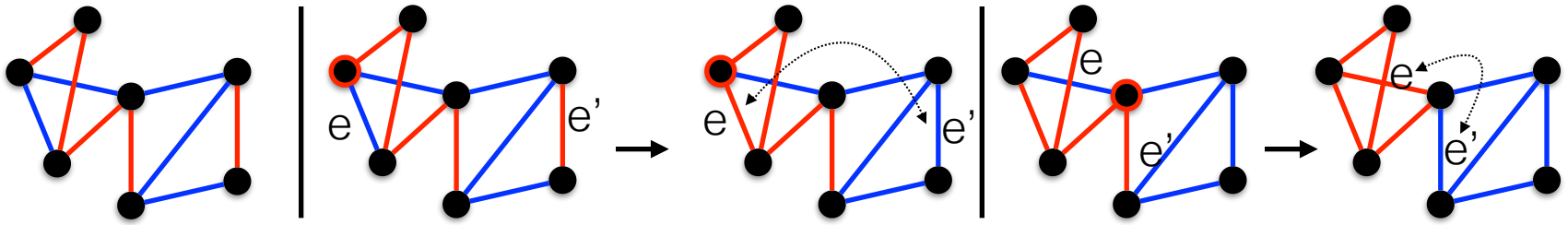
Gain

Heat

$$v(e,c') = \frac{\sum_{C_i \in N_e, C_{i=}c'} C_i}{|C_{N_e}|}$$

Vertex-Cut Partitioning

k = 2



Random Initialization

Iteration

Iteration

C = Blue
C' = Red

$$utility = ((v(e, c') + v(e', c)) \times T_r) - (v(e, c) + v(e', c'))$$

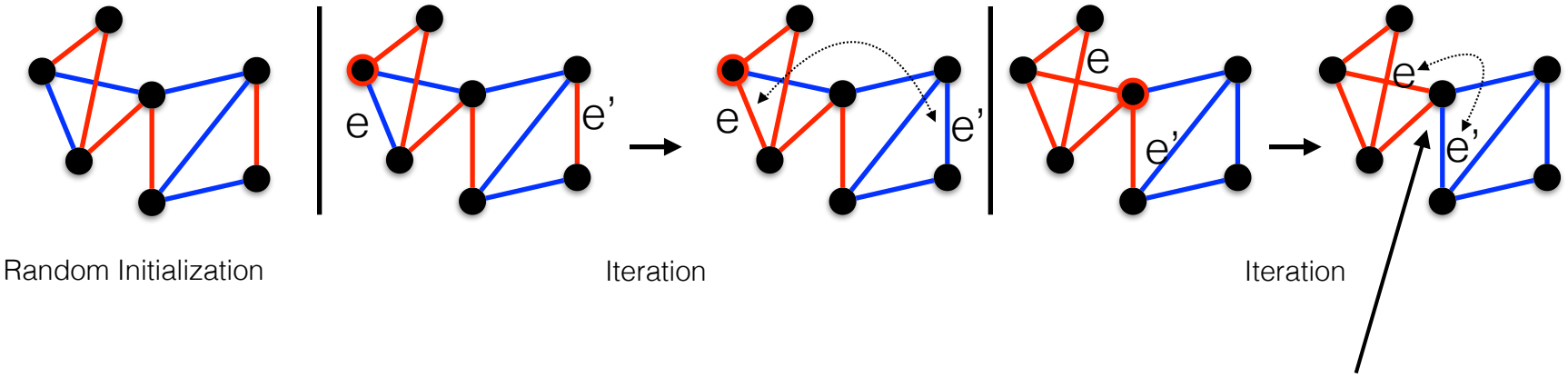
Gain

Heat

$$v(e, c') = \frac{\sum_{C_i \in N_e, C_{i=c'}} C_i}{|C_{N_e}|}$$

Vertex-Cut Partitioning

k = 2



C = Blue
C' = Red

$$utility = ((v(e, c') + v(e', c)) \times T_r) - (v(e, c) + v(e', c'))$$

Gain Heat

$$v(e, c') = \frac{\sum_{C_i \in N_e, C_{i=c'}} C_i}{|C_{N_e}|}$$

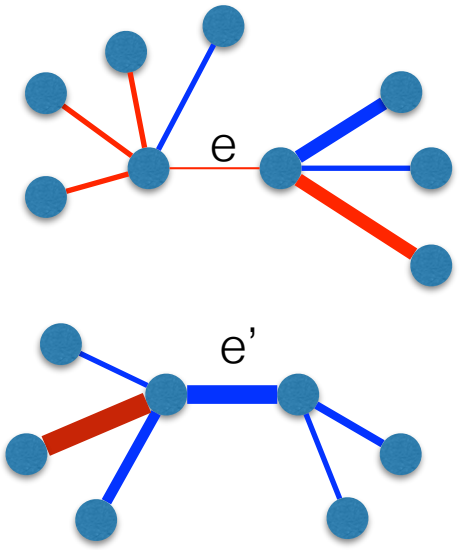
Modifications

- Utility Function

$$utility = ((v(e,c') + v(e',c)) \times T_r) - (v(e,c) + v(e',c'))$$

- Weighted Graph

$$v(e,c') = \frac{\sum_{C_i \in N_e, C_{i=}c'} C_i}{|C_{N_e}|} \rightarrow v(e,c') = \frac{\sum_{C_i \in N_e, C_{i=}c'} W C_i}{\sum_{i \epsilon N_e} W_i}$$



- Balance Threshold

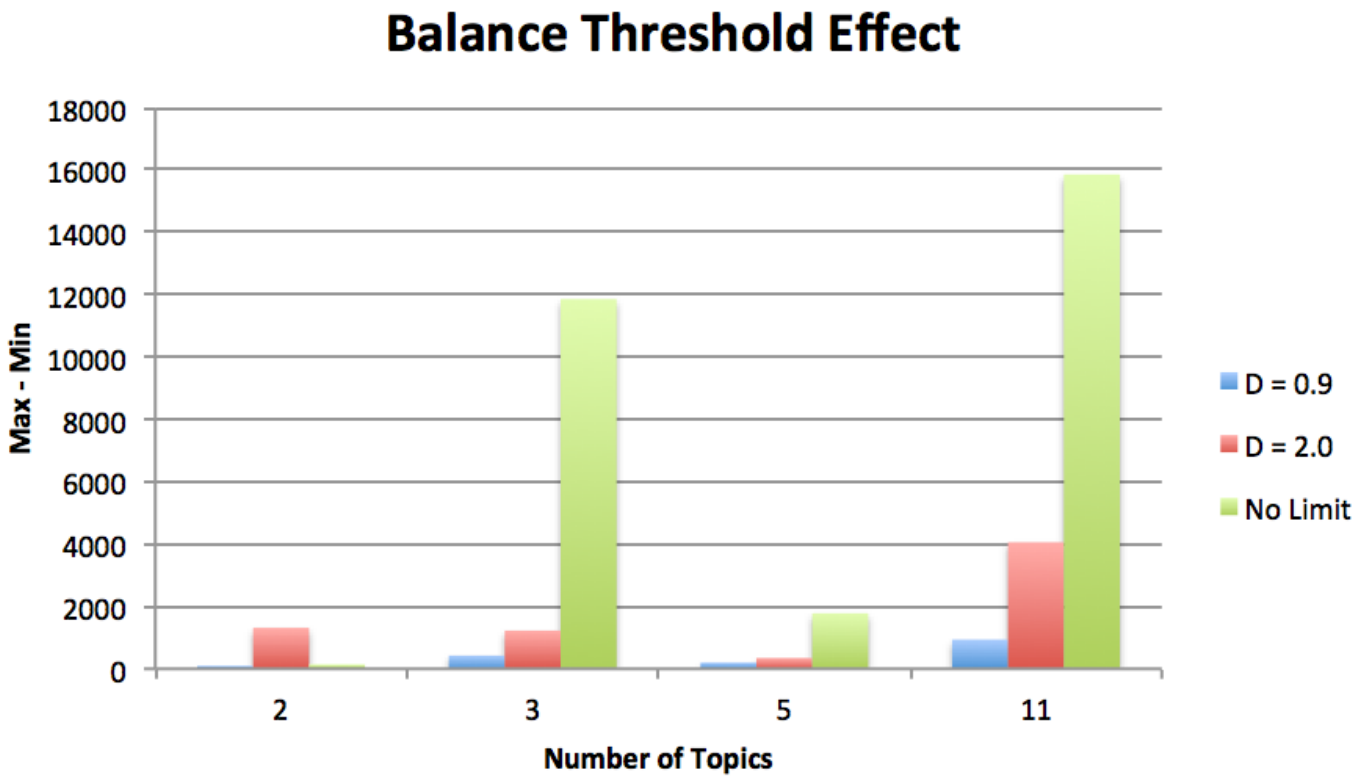
$$|w_e - w_{e'}| < \delta$$

Experiments

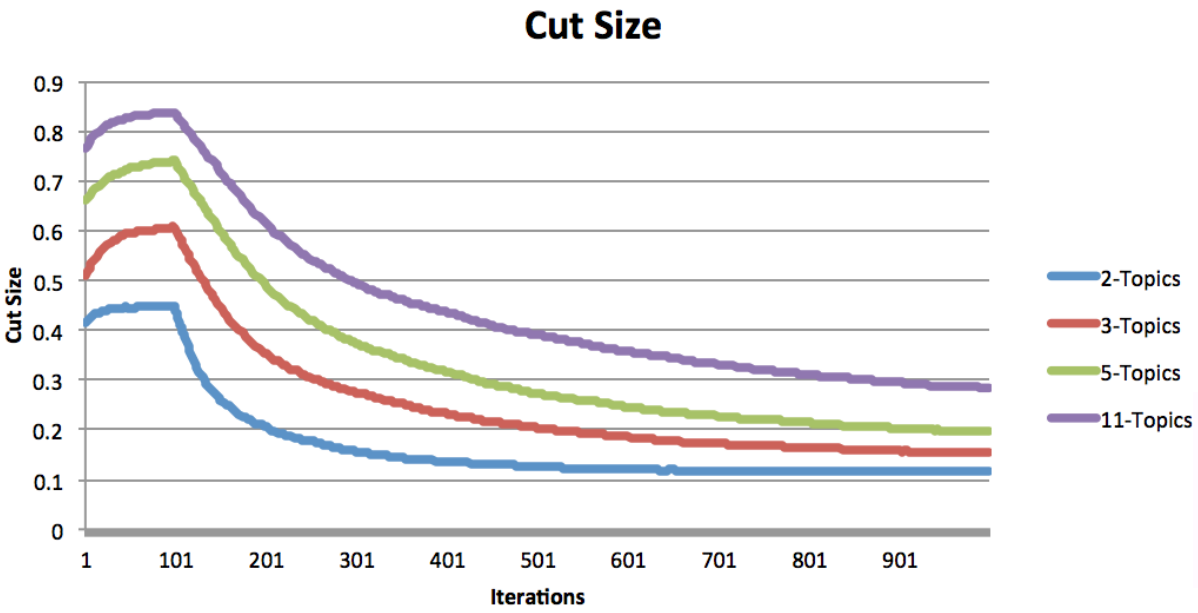
- **Data set**
  - SNAP - Tweets - 2009
  - Trending Topics
- **Uniform Distribution**
  - 2 topics
  - 3 Topics
  - 5 Topics
- **Skewed Distribution**
  - 11 topics
- **RI - Graph**
  - 2000 Vertices
  - 1.98m Edges

News Events	Movies	Sports (Teams, Events, Leagues)
1. #iranelection	1. Harry Potter	1. Super Bowl
2. Swine Flu	2. New Moon	2. Lakers
3. Gaza	3. District 9	3. Wimbledon
4. Iran	4. Paranormal Activity	4. Cavs (Cleveland Cavaliers)
5. Tehran	5. Star Trek	5. Superbowl
6. #swineflu	6. True Blood	6. Chelsea
7. AIG	7. Transformers 2	7. NFL
8. #uksnow	8. Watchmen	8. UFC 100
9. Earth Hour	9. Slumdog Millionaire	9. Yankees
10. #inaug09	10. G.I. Joe	10. Liverpool
People	**TV Shows**	**Technology**
1. Michael Jackson	1. American Idol	1. Google Wave
2. Susan Boyle	2. Glee	2. Snow Leopard
3. Adam Lambert	3. Teen Choice Awards	3. Tweetdeck
4. Kobe (Bryant)	4. SNL (Saturday Night Live)	4. Windows 7
5. Chris Brown	5. Dollhouse	5. CES
6. Chuck Norris	6. Grey's Anatomy	6. Palm Pre
7. Joe Wilson	7. VMAS (Video Music Awards)	7. Google Latitude
8. Tiger Woods	8. #bsg (Battlestar Galatica)	8. #E3
9. Christian Bale	9. BET Awards	9. #amazonfail
10. A-Rod (Alex Rodriguez)	10. Lost	10. Macworld

Discussion


Balance Threshold Effect
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Max - Min
2
3
5
11
Number of Topics
D = 0.9
D = 2.0
No Limit

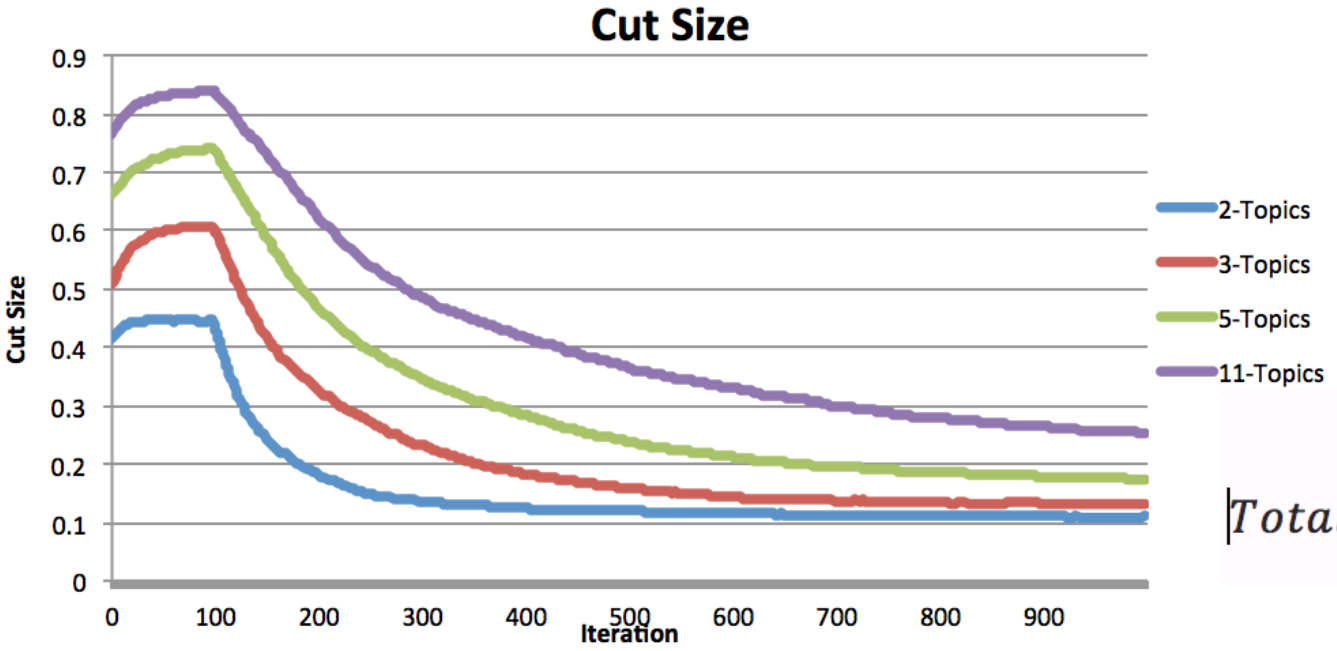
Discussion



$$Total\ Cut = 1 - \frac{\sum_{v \in V} \left(\frac{|W_D|}{|W|}\right)_v}{|V|}$$

Num Topics	Size	F-Score
2	2715	0.59
3	4169	0.66
5	8326	0.33
11	31231	0.42

Discussion

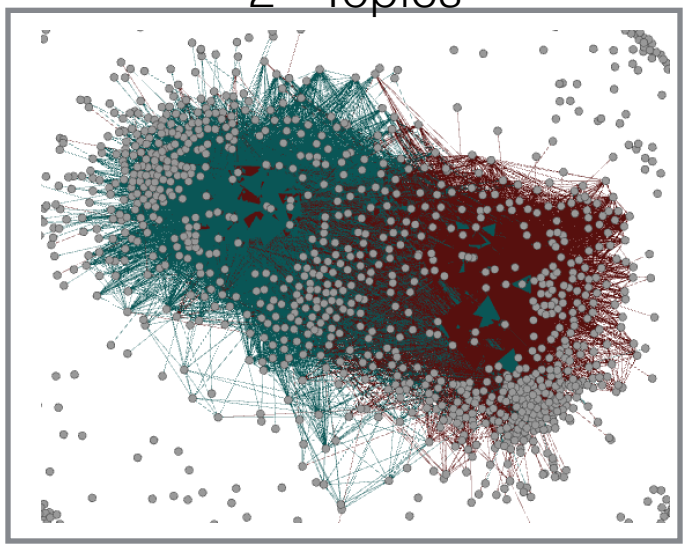


$$Total\ Cut = 1 - \frac{\sum_{v \in V} \left(\frac{|W_D|}{|W|}\right)_v}{|V|}$$

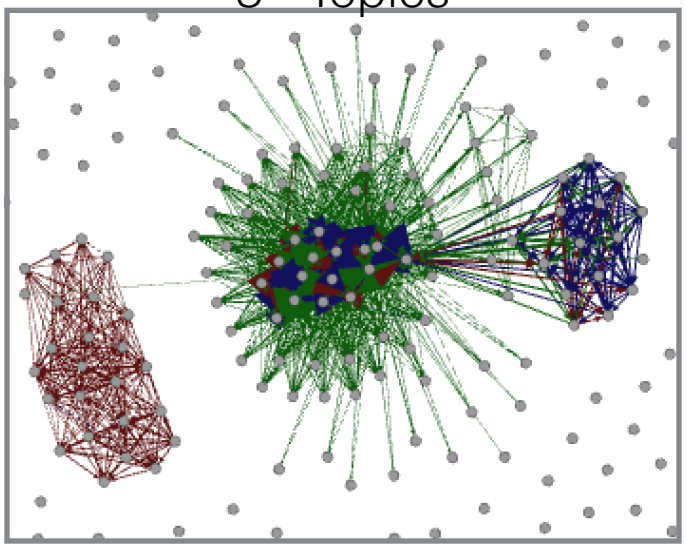
Num Topics	Size	F-Score
2	2715	**0.98**
3	4169	**0.92**
5	8326	**0.62**
11	31231	0.36

Discussion

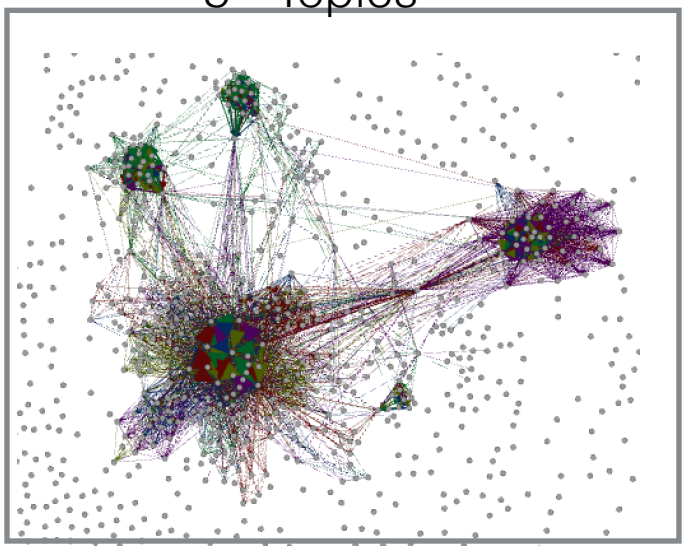
2 - Topics



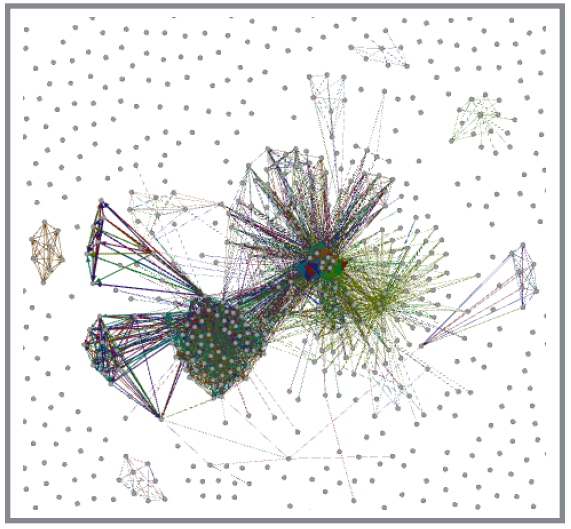
3 - Topics



5 - Topics



11 - Topics



Conclusion

- Advantage
  - Scalable
  - Incremental
  - Uniform Partition Distribution
- Challenge
  - Non-Uniform Partition Distribution
- Future work
  - Global Shared partition size monitor
  - Enhance Initialization (BFS vs Random)
  - Split Weights

Thank You

Questions?

Bibliography

1. Sahlgren, M. (2005) An Introduction to Random Indexing, Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005, August 16, Copenhagen, Denmark.
2. Kanevara, P: Sparse Distributed Memory and Related Models. Associative Neural Memories, Oxford University Press, 1993.
3. Kanerava, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In Gleitman, L. R. and Josh, A. K., editors, Proceedings of the 22nd Annual Conference of the Cognitive Science Society, page 1036, Mahwah, New Jersey. Erlbaum.
4. Johnson, W. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hil- bert space. In Beals, R., Beck, A., Bellow, A., and Hajian, A., editors, Conference on Modern Analysis and Probability (1982: Yale University), volume 26 of Con- temporary Mathematics, pages 189–206. American Mathematical Society.
5. K Ghoorchian, F Rahimian, S Girdzijauskas: Semi Supervised Multiple Disambiguation, Trustcom/BigDataSE/ISPA, 2015 IEEE 2, 88-95.

img

1. Img 1 - http://www.studerasmart.nu/wp-content/uploads/2012/04/jobb-och-cv.png
2. Img 2 - http://gfx2.aftonbladet-cdn.se/image/19456728/485/normal/efc46e3660c6c/hedenmo3.jpg
3. Img 3 - http://cdn01.nyheter24.se/c4ab6c0402fa00a700/2014/04/03/941973/Sk%C3%A4rmavbild%202014-04-03%20kl.%2020.54.47.png
4. Img 4 - http://ericagelfandlaw.com/wp-content/uploads/2015/12/immigration.jpg