Link Classification and Clustering in Signed Graphs

Géraud Le Falher
Inria Lille – Nord Europe, Magnet, France

What it is a signed graph?

- Graphs model relationships between entities
- **homophily** assumption: linked entities tend to share common properties
- the strength of these relations is quantified by weights $w_{u,v} \in \mathbb{R}^+$
- allowing weights to be negative gives new semantics to edges:
  - dissimilarity
  - distrust
  - enmity

Problems and applications

1. Link classification: **predict the signs of a set of edges**, given the graph structure and the signs of the other edges. Applications in:
   - understanding trust dynamic and communities formation
   - testing social theories at a large scale
   - **recommending products**

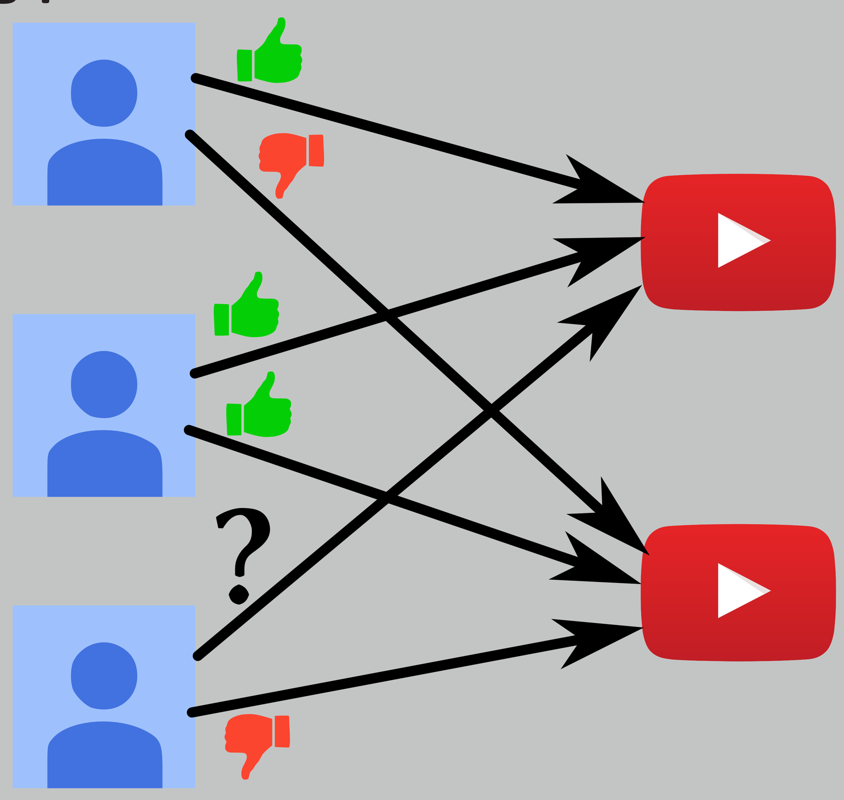


Figure: Youtube users/votes/videos bipartite signed graph

2. Correlation Clustering: **find a partition of the nodes minimizing the number of disagreement edges** (i.e. positive edges between clusters and negative edges within clusters). Used in:
   - image segmentation
   - solving co-reference task in natural language processing
   - identifying genes relationship
   - entity resolution
   - aggregation of clusterings

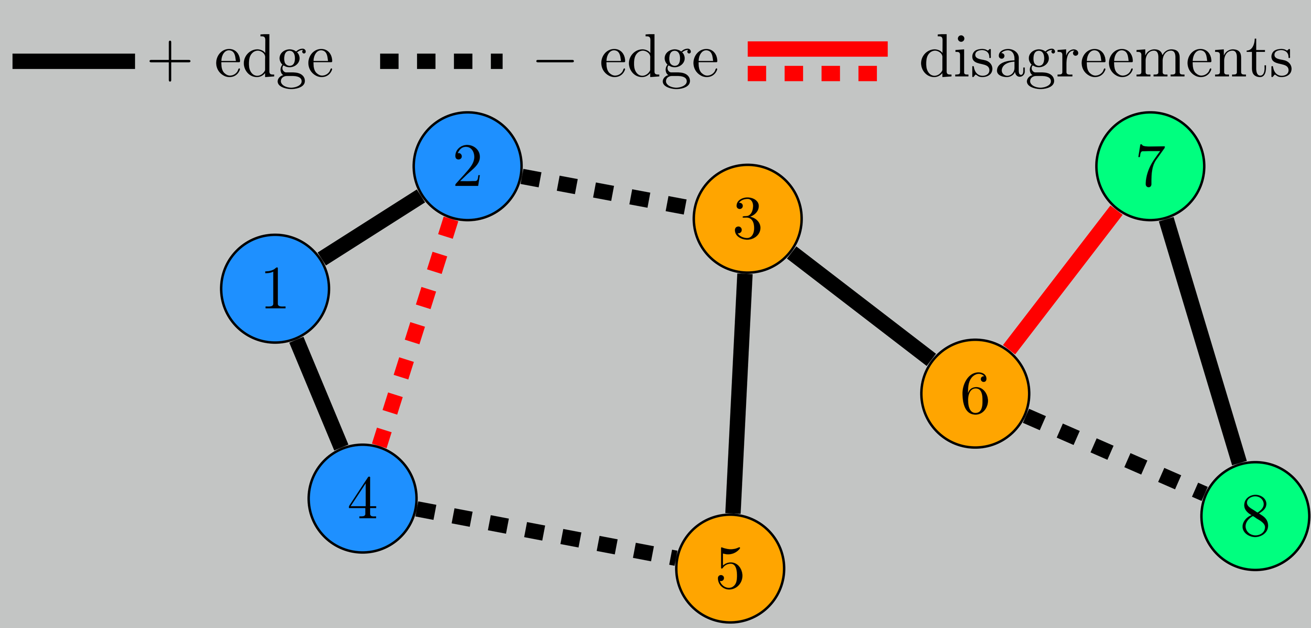


Figure: The optimal clustering of this signed graph has a cost of 2.

Link classification: current approaches

- Batch setting
  - train logistic regression on triangle patterns [7]
  - train logistic regression on higher order cycles [3]
  - train SVM on frequent subgraphs [9]

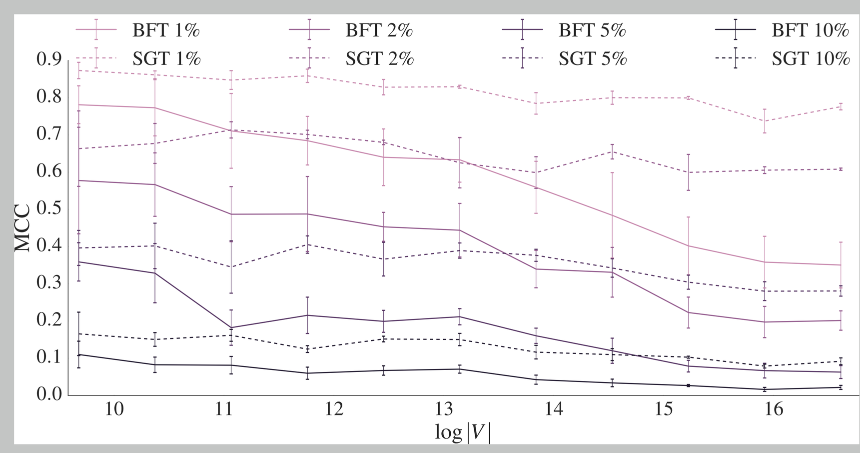
  Achieve good generalization but require a large training set
- Active setting: Parsimoniously select a training set to minimize the prediction error on the testing set [2]

Link classification: our goals

- Devise efficient graph sparsifiers T
  - In the p-stochastic two clusters case, all the signs of T are queried
  - For any $(u, v) \in E$, predict $\hat{y}_{u,v} = \prod_{e \in path^T_{u,v}} y_e$
  - Thus we want $path^T_{u,v}$ to be short for all $(u, v) \in E$
- Be adaptive, i.e. select at each step the edges adding the more information given those already revealed

Link classification: preliminary results

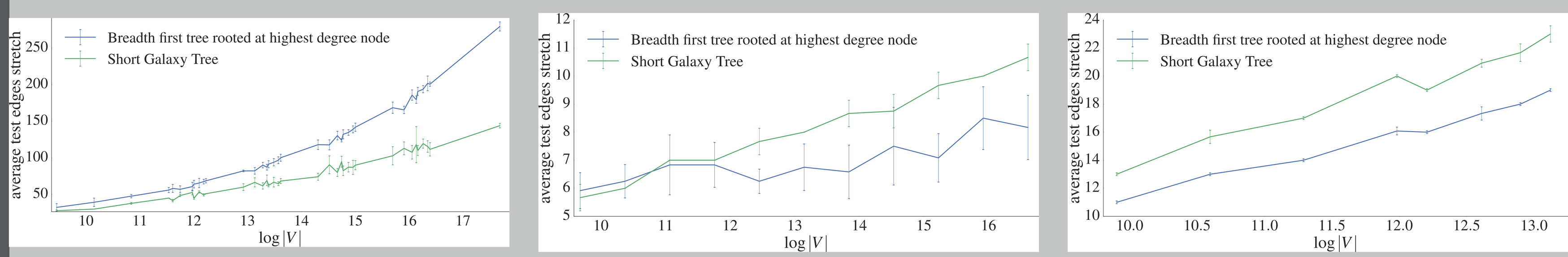
- T is a spanning tree of G
- $E_{test} = E \setminus T$
- $stretch = \frac{1}{|E_{test}|} \sum_{(u,v) \in E_{test}} |path^T_{u,v}|$
- The lower the stretch, the better the prediction
- Use Matthews Correlation Coefficient to assess binary prediction



$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} = \pm\sqrt{\frac{\chi^2}{n}}$$

Table: MCC on 3 real datasets. Our method is more robust, albeit slightly less accurate

	Wikipedia	Slashdot	Epinion
$\lvert V \rvert$	7 065	82 052	119 070
$\lvert E \rvert$	99 936	498 527	701 569
fraction of + edges	78.5%	76.4%	83.2%
Breadth First Tree	.185 (.075)	.159 (.074)	.255 (.115)
Our Tree	.164 (.045)	.169 (.028)	.216 (.030)



Grid Pref. Attachment Triangle

Correlation Clustering: current approaches

- APX-hard problem
- On complete graphs, KwikCluster algorithm provides a simple and efficient combinatorial 3-approximation [1]

  function KwikCluster($G = (V, E)$)
    while not all nodes are clustered **do**
      $pivot \leftarrow$ pick a node in V at random
      put $pivot$ in its own cluster
      add all its positive neighbors
      remove them from G
- On general graphs, the approximation ratio is $O(\log n)$ and is obtained by solving a large SDP.

Correlation Clustering: our goals

- A combinatorial algorithm for general graphs, retaining the optimal $O(\log n)$ approximation ratio
- An idea would be to complete the graph through simple rules drawn from strong balance theory [5]
- Address the scalability issue, by studying parallel [8] and distributed (MapReduce or Pregel-like framework) [4] versions of KwikCluster, as well as parallelized BOEM post processing [6]

Correlation Clustering: preliminary results

- Planted model: k clusters of roughly n nodes each and consistent edge signs
- Flip the signs of 7% edges and assume it is the optimal number of errors ϕ

Table: The ratio of our number of mistakes to ϕ is constant

k	5	2	30	2	20	10	15
n_i	15	65	6	100	12	25	35
nodes	75	130	180	200	240	250	525
ratio	2.2	2.6	1.6	3.0	1.7	2.1	1.9

References

1. Ailon et al. 2008, Aggregating inconsistent information. Journal of the ACM, 55(5)
2. Cesa-Bianchi et al. 2012, A linear time active learning algorithm for link classification. NIPS'12
3. Chiang et al. 2014, Prediction and Clustering in Signed Networks: A Local to Global Perspective. JMLR
4. Chierichetti et al. 2014, Correlation clustering in MapReduce. KDD'14
5. Davis, 1967, Clustering and structural balance in graphs. Human Relations, 20
6. Elsner et al. 2009, Bounding and comparing methods for correlation clustering beyond ILP, NAACL HLT'09
7. Leskovec et al., 2010, Predicting positive and negative links in online social networks. WWW'10
8. Pan et al., 2014, Scaling up Correlation Clustering through Parallelism and Concurrency Control. NIPS'14 Workshop
9. Papaoikonomou et al. 2014, Predicting Edge Signs in Social Networks Using Frequent Subgraph Discovery. IEEE Internet Computing, 18(5)

 geraud.le-falher@inria.fr