Distributed and Parallel algorithms for Correlation Clustering

context

In the Correlation Clustering [Reference 1] (CC) problem, the input is a graph whose edges carry a *sign* representing the positive (similarity) or negative (dissimilarity) nature of the relationship between the incident nodes. These graphs are called *signed graphs*.

For instance, disapproval or distrust in social networks, negative endorsements on the Web. Concrete examples are provided by certain types of online social networks [Reference 2]. Users of Slashdot can tag other users as friends or foes. Similarly, users of Epinions can give positive or negative ratings not only to products but also to other users. Even in the social network of Wikipedia administrators, votes cast by an admin in favor or against the promotion of another admin can be viewed as positive or negative links. More examples of signed links are found in other domains, such as the excitatory or inhibitory interactions between genes or gene products in biological networks.

The goal of CC is then to cluster the nodes of an input signed graph according the given similarity information. Formally, given any node partition into clusters, a *disagreements edge* is a negative edge linking two nodes within the same cluster or a positive edge between different clusters. Thus, the goal is to find a partition minimizing the number of such disagreement edges.

Though CC is NP-hard, there exist a randomized combinatorial algorithm due to Ailon et al. [Reference 3] that is, in expectation, optimal up to a factor 3 when the input signed graph is an unweighted clique.

As current graphs can be very large, scalability become an issue and require new approaches. For instance parallel computation, which takes advantage of the multi cores of modern hardware architecture [Reference 4]. When the graph is too large to fit in a single machine, it is possible to partition it and use formalisms such as MapReduce or Bulk Synchronous Parallel model to compute a solution in several rounds of message passing [Reference 5].

missions

- getting familiar with Correlation Clustering problem
- realizing a state of the art regarding Distributed and Parallel approximation on general graph
- implementing one of the algorithm from this literature review
- studying its performance on synthetic on real data

references

[1]N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pp. 238–247, 2002 [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1181947

[2]J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proceedings of the 19th international conference on World wide web - WWW '10*, 2010, p. 641 [Online]. Available: http://portal.acm.org/citation.cfm?doid=1772690.1772756

[3]N. Ailon, M. Charikar, and A. Newman, "Aggregating inconsistent information," *Journal of the ACM*, vol. 55, no. 5, pp. 1–27, Oct. 2008 [Online]. Available: http://portal.acm.org/citation.cfm?doid=1411509.1411513

[4]X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. I. Jordan, "Scaling up Correlation Clustering through Parallelism and Concurrency Control," in *NIPS Workshop on Discrete and Combinatorial Problems in Machine Learning (DISCML)*, 2014 [Online]. Available: http://las.ethz.ch/discml/final14/20/_Pan/_scaling.pdf

[5]F. Chierichetti, N. Dalvi, and R. Kumar, "Correlation clustering in MapReduce," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, 2014, pp. 641–650 [Online]. Available: http://dl.acm.org/citation.cfm?id=2623330.2623743