Correlation Clustering with Adaptive Similarity Queries

Marco Bressan
Nicolò Cesa-Bianchi
Andrea Paudice
Fabio Vitale

(Query-Based) Correlation Clustering

Let $V = [n]$ be a set of elements and $\sigma : V \times V \to \{-1, +1\}$ be a similarity function. Given V and a budget Q of queries to σ, the goal in Correlation Clustering (CC) is to find a clustering $\mathcal{C}$ of V minimizing the clustering error

$$\Delta_{\mathcal{C}} = \sum_{\mathcal{C}(u)=\mathcal{C}(v)} \mathbb{I}\big(\sigma(u,v) = -1\big) + \sum_{\mathcal{C}(u)\neq\mathcal{C}(v)} \mathbb{I}\big(\sigma(u,v) = +1\big) \ .$$

OPT denotes the cost of an optimal solution to CC.

Remark 1: The number of clusters to use is not fixed *a priori*.
Remark 2: CC is NP-Hard and hard to approximate even in the case σ is fully known beforehand.

Information Theoretic Bounds

Idea: CC can be seen as binary classification under uniform marginal and using cluster-based classifiers!

- Upper Bound: Using VC theory we show that the combination of Uniform Sampling with Empirical Risk Minimization (USERM) leads to an algorithm that with Q queries outputs (with high probability) a clustering with cost of $\mathrm{OPT} + O\left(\frac{n^{5/2}}{\sqrt{Q}}\right)$.
- Lower Bound: Any (possibly randomized and adaptive) algorithm asking Q queries to σ is forced to output a clustering with (expected) cost of $\mathrm{OPT} + \Omega\left(\frac{n^3}{Q}\right)$.

Remark 1: USERM is a fully additive algorithm and therefore must necessarily run in exponential time.
Remark 2: Even when $\mathrm{OPT} = 0$, any algorithm that asks Q queries must output a clustering with (expected) cost of $\Omega\left(\frac{n^2}{\sqrt{Q}}\right)$.

Cluster Recovery

Beyond small clustering error, ACC it is also able to recover ground truth clusters that are close enough to a clique.

A subset C of V is called $(1-\epsilon)$-knit if $|E_C| \geqslant (1-\epsilon)\binom{|C|}{2}$ and $|\mathrm{cut}(C, C')| \leqslant \epsilon\binom{|C|}{2}$. C is called strongly $(1-\epsilon)$-knit if $|E_C| \geqslant (1-\epsilon)\binom{|C|}{2}$ and $|\mathrm{cut}(C, C')| = 0$.

For any $(1-\epsilon)$-knit, ACC finds a cluster $\widehat{C}$ such that $\mathbb{E}\big[C \Delta \widehat{C}\big] = O\big(\epsilon|C| + n^2/Q\big)$. With $O\big(\ln(n)\big)$ repetitions of ACC, all strongly $(1-\epsilon)$-knit clusters can be recovered with high probability by assigning clusters to nodes via majority voting.

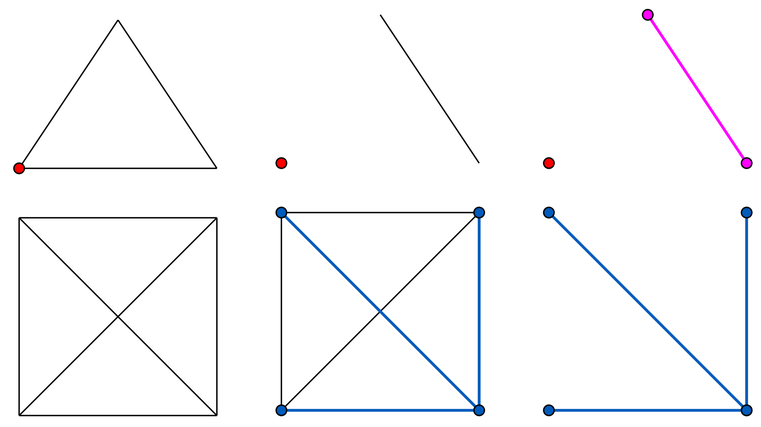
Remark: An $(1-\epsilon)$-knit cluster can be even generated by an adversary which starts with a clique and then arbitrary corrupts a fraction ϵ of its edges.

ACC

Active Correlation Clustering (ACC) is a simple variant of KwikCluster that with a budget of Q queries to σ outputs a clustering with expected cost of $3\mathrm{OPT} + O\left(\frac{n^3}{Q}\right)$.

High level description of ACC:

- Set $f(n) = o(n)$.
- Run for $f(n)$ rounds.
- In each round r select u.a.r. a pivot π_r from the residual graph V_r, choose u.a.r. $f(n_r)$ elements in V_r/π_r, and ask σ for their similarity with the pivot.
- If no positive similarities are observed, declare π_r singleton, otherwise ask for the remaining similarities and form the cluster with π_r and its neighbors.
- After $f(n)$, declare any remaining node in $V_{f(n)}$ singleton.



Remark: Note that the actual input of ACC is the query rate $f(n)$ and it performs at most $nf(n)$ queries.

ACCESS

Idea: The number of queries can be reduced by testing the size of the residual graph!

Active Correlation Clustering with Early Stop Strategy (ACCESS), in each round, tests the number of edges of the residual graph. If a small number of edges is left, it stops and declares the remaining nodes singleton, otherwise it performs a round of ACC. ACCESS has the same expected cost of ACC and has a guarantee on the expected number of queries. However, on some graph ACCESS performs much less queries.

Example: Let $f(n) = \sqrt{n}$ and supose G contains $n^{1/3}$ cliques of $n^{2/3}$ nodes each. Then ACC performs $O\big(n^{3/2}\big)$ queries, while ACCESS performs only $O\big(n^{4/3}\ln(n)\big)$ queires.

Experimental Results