Active clustering: partitioning using pairwise queries

Quentin Lutz, Élie de Panafieu, Alex Scott, Maya Stein

Nokia Bell Labs, Oxford University, University of Chile
Randnet project

AofA 2021

Problem

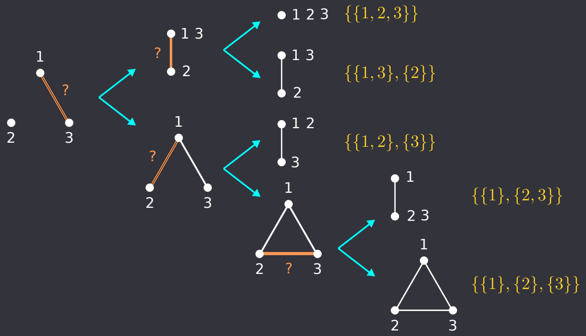
Sorting problem. Recover an unknown **permutation** $(3, 1, 4, 5, 2)$ using pairwise queries "$a < b$?".

Active clustering. Recover an unknown **set partition** $\{\{1, 3\}, \{2, 5\}, \{4\}\}$ using pairwise queries "$a \sim b$?".

In both cases, the complexity is the number of queries.

We have not found this very natural setting in the literature, any suggestion would be greatly appreciated.

Example



Transitivity. If $a \sim b$ then $\text{query}(a, c) = \text{query}(b, c)$

Aggregated graph	vertex	=	set of similar elements
	edge	=	dissimilar groups of elements

Answer to a query (u, v)	positive	$\rightarrow$	merge u and v
	negative	$\rightarrow$	add an edge (u, v)

Structure of the talk

Theorem 1. Characterize the active clustering algorithms reaching the minimal average complexity.

Theorem 2. Those algorithms share the same complexity distribution.

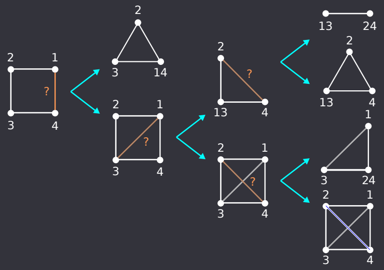
Theorem 3. Characterize this distribution and prove a Gaussian limit law.

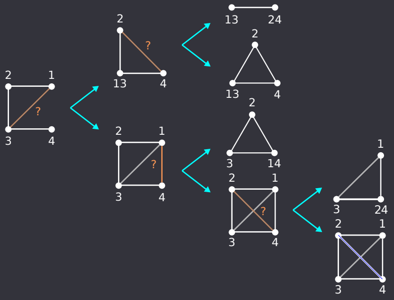
Motivation. From Maria Laura Maag (Nokia), improve the classification of training data by human experts to feed a supervised learning software.

We assume uniform distribution on the partitions.

Initial conjecture. All non-trivial queries lead to the same average complexity.

Counter-example. Average complexity $\frac{13}{5} \neq \frac{12}{5}$.





Unexpected answer

Theorem 1. An active clustering algorithm has minimal average complexity iff all aggregated graphs are chordal.

Induced graph. The graph  is induced in  but not in .

Chordal graph. A graph is chordal if all induced cycles are triangles.

chordal  ; non-chordal 

Chordal query (u, v). The intersection of the neighborhoods $\mathcal{N}(u) \cap \mathcal{N}(v)$ separates u and v.

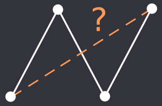
query $(1, 3)$ is chordal, while $(1, 4)$ is not.

Proof of Theorem 1

If the partition contains n elements and k blocks $B_1, \ldots, B_k$, then

$$\begin{aligned} \text{number of positive answers} &= n - k \\ \text{number of negative answers} &= \textstyle\sum_{i<j} \#\ \text{queries between } B_i \text{ and } B_j \end{aligned}$$

If we know there are 2 blocks, then queries between ends of odd induced paths are wasteful (automatic negative answer)

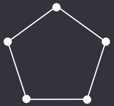


An algorithm has minimal average complexity on 2 blocks iff it avoids wasteful queries.

Proof of Theorem 1

Chordal algorithms avoid potential wasteful queries for all subsets that could be the union of two blocks. Indeed, bipartite induced subgraphs of chordal graphs are forests.

Non-chordal algorithms contain at least one wasteful query. Consider the aggregated graph after a negative answer to the first query that creates an induced cycle C of length ≥ 4. If $|C|$ is even, the query was wasteful. Otherwise, the first query inside C will be wasteful.



Theorem 2

Theorem 1. An active clustering algorithm has minimal average complexity iff it is chordal.

Theorem 2. All chordal algorithms have the same complexity distribution.

Proof. By induction on the number of missing edges of the aggregated graph, we prove that all chordal queries give the same complexity distribution.

Initialization. If G is a complete graph, there are no more queries to ask.

Proof of Theorem 2

Notations. $G(u, v) =$ add edge, $G(u/v) =$ merge vertices.
If G and $G(u, v)$ are chordal, then so is $G(u/v)$.

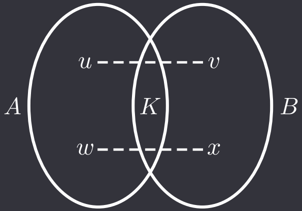
The complexity distribution is the height distribution of the leaves of the query tree.

Induction. Consider two chordal queries (u, v) and (w, x). If $G(u, v)(w, x)$ is chordal, then the queries can be switched

$$\begin{tikzcd}
 & G(u/v) \arrow[r] \arrow[dr] & G(u/v)(w/x) \\
G \arrow[ur] \arrow[dr] & & G(u/v)(w,x) \\
 & G(u,v) \arrow[r] \arrow[dr] & G(u,v)(w/x) \\
 & & G(u,v)(w,x)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & G(w/x) \arrow[r] \arrow[dr] & G(w/x)(u/v) \\
G \arrow[ur] \arrow[dr] & & G(w/x)(u,v) \\
 & G(w,x) \arrow[r] \arrow[dr] & G(w,x)(u/v) \\
 & & G(w,x)(u,v)
\end{tikzcd}$$

Proof of Theorem 2

Otherwise, G, $G(u,v)$, $G(w,x)$ are chordal, $G(u,v)(w,x)$ is not. This constrains the structure of G



Asking (u,v) or (w,x), then turning A and B into cliques lead to two symmetrical situations, so the complexity distributions are the same.

Theorem 3

Bell number	B_n	$=$	number of set partitions of size n
Lambert function	$W(x)$	$=$	solution of $we^w = x$
q-analog	$[n]_q$	$=$	$1 + q + \cdots + q^{n-1}$
q-factorial	$[n]_q!$	$=$	$[1]_q \times [2]_q \times \cdots \times [n]_q$
q-exponential	$e_q(z)$	$=$	$\sum_{n \geq 0} \frac{z^n}{[n]_q!}$
q-Pochhammer	$(a; q)_n$	$=$	$(1 - aq^0) \times \cdots \times (1 - aq^{n-1})$

Theorem 3. Let X_n denote the complexity of a chordal algorithm on a partition of size n chosen uniformly at random.
The probability generating function (PGF) of X_n is equal to

$$\frac{1}{B_n} \left(\frac{q}{1-q} \right)^n \sum_{k=0}^{n} \binom{n}{k} (-1)^k \left(\frac{1-q}{q}; q \right)_k \quad \text{and} \quad \frac{1}{B_n} \frac{1}{e_q(1/q)} \sum_{m \geq 0} \frac{[m]_q^n}{[m]_q!} q^{n-m}.$$

The normalized variable $(X_n - E_n)/\sigma_n$ converges in distribution to a standard Gaussian law, where

$$E_n = \frac{1}{4}(2W(n) - 1)e^{2W(n)} \quad \text{and} \quad \sigma_n = \frac{1}{3}\sqrt{\frac{3W(n)^2 - 4W(n) + 2}{W(n) + 1} e^{3W(n)}}.$$

A q-analog of Bell numbers

The generating function $P(z)$ of set partitions is

$$P(z) = \mathrm{Set}(\mathrm{NonEmptySet}(z)) = e^{e^z - 1}$$

so the nth Bell number is

$$B_n = n![z^n]P(z) = \frac{n!}{e}[z^n] \sum_{m \geq 0} \frac{e^{mz}}{m!} = \frac{1}{e} \sum_{m \geq 0} \frac{m^n}{m!}$$

Our second formula for the complexity GF is a q-analog

$$\frac{1}{e_q(1/q)} \sum_{m \geq 0} \frac{[m]_q^n}{[m]_q!} q^{n-m}.$$

Universal active clustering algorithm

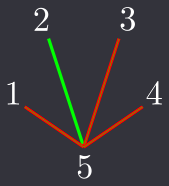
Theorem 1. The active clustering algorithms with minimal average complexity are the chordal algorithms.

Theorem 2. All chordal algorithms share the same complexity distribution.

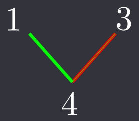
Thus, we analyze a particular case: the universal active clustering (UAC) algorithm.

```
def UAC(S):
    if S is empty:
        return empty partition
    else:
        u = S.pop()
        query u with all elements from S
        B = block containing u
        Q = UAC(S \ B)
        return partition Q with an additional block B
```

Example of UAC execution



$\{\{2, 5\}\}$



$\{\{1, 4\}, \{2, 5\}\}$

3

$\{\{3\}, \{1, 4\}, \{2, 5\}\}$

Generating function of UAC complexity

Generating function of set partitions $P(z) = \sum_{\text{partition } p} \frac{z^{|p|}}{|p|!}$

Bijection

partition (not counting the largest label) pair (partition, set)

$\{\{1,3\},\{4\},\{2,5,6\}\}$ $(\{\{1,3\},\{4\}\},\ \{2,5\})$

Symbolic method $\partial_z P(z) = P(z) e^z$

(no surprise, as $P(z) = e^{e^z - 1}$).

Additional variable q marking the queries used by UAC

$$P(z,q) = \sum_{\text{partition } p} q^{\text{queries}(p)} \frac{z^{|p|}}{|p|!}, \quad \partial_z P(z,q) = P(qz,q) e^{qz}$$

Solving the differential equation

Since $f(z,q) = e^{\frac{q}{1-q}z}$ satisfies the similar diff eq

$$\partial_z f(z,q) = \frac{q}{1-q} f(qz,z) e^{qz}$$

we search solutions of the form $P(z,q) = A(z,q) f(z,q)$.

Diff eq on $P(z,w)$ $\rightarrow$ diff eq on $A(z,q)$ $\rightarrow$ recurrence on its Taylor coefficients

$$A(z,q) = \sum_{k \geq 0} \left(\frac{1-q}{q}; q \right)_k \frac{\left(-\frac{q}{1-q} z \right)^k}{k!}$$

Exact expressions

We obtain by direct coefficient extraction

$$n![z^n]P(z,q) = \left(\frac{q}{1-q}\right)^n \sum_{k=0}^{n} \binom{n}{k} (-1)^k \left(\frac{1-q}{q}; q\right)_k .$$

To prove the second expression

$$n![z^n]P(z,q) = \frac{1}{e_q(1/q)} \sum_{m\geq 0} \frac{[m]_q^n}{[m]_q!} q^{n-m},$$

we apply the classic q-identities

$$[n]_q! = \frac{(q;q)_n}{(1-q)^n}, \qquad \frac{1}{(x;q)_\infty} = \sum_{n\geq 0} \frac{x^n}{(q;q)_n}, \qquad e_q(x) = ((1-q)x;q)_\infty^{-1}.$$

Limit law

To obtain the Gaussian limit law, we prove that the Laplace transform of the normalized random variable $X_n^\star = (X_n - E_n)/\sigma_n$

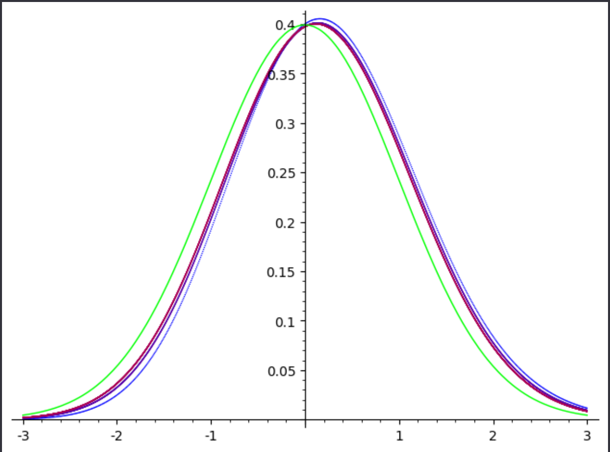
$$\mathbb{E}(e^{sX_n^\star}) = \mathrm{PGF}_n(e^{s/\sigma_n})e^{-sE_n/\sigma_n}$$

converges to the Laplace transform of the standard Gaussian $e^{s^2/2}$ pointwise for s in a neighborhood of 0.

To do so, we apply the Laplace method for sums to

$$\mathrm{PGF}_n(e^{s/\sigma_n}) = \left[\frac{1}{B_n}\frac{1}{e_q(1/q)}\sum_{m\geq 0}\frac{[m]_q^n}{[m]_q!}q^{n-m}\right]_{q=e^{s/\sigma_n}}$$

Local limit law



Green: probability density function of the standard normal law.
Blue, purple and red: empirical rescaled probability density functions of chordal complexity for $n \in \{100, 300, 600\}$.

Conclusion

Open problem. Complexity of a random active clustering algorithm avoiding trivial queries?

Other random partition model. Fix a bound k on the number of blocks, each item chooses block i with probability p_i.

Results. Average complexities of the conjectured best algorithm and the random algorithm.

Noisy queries. Two models

- correct at most k errors,
- small probability p of error for each answer; minimize the probability of undetected errors.