OR / Data science in online advertising

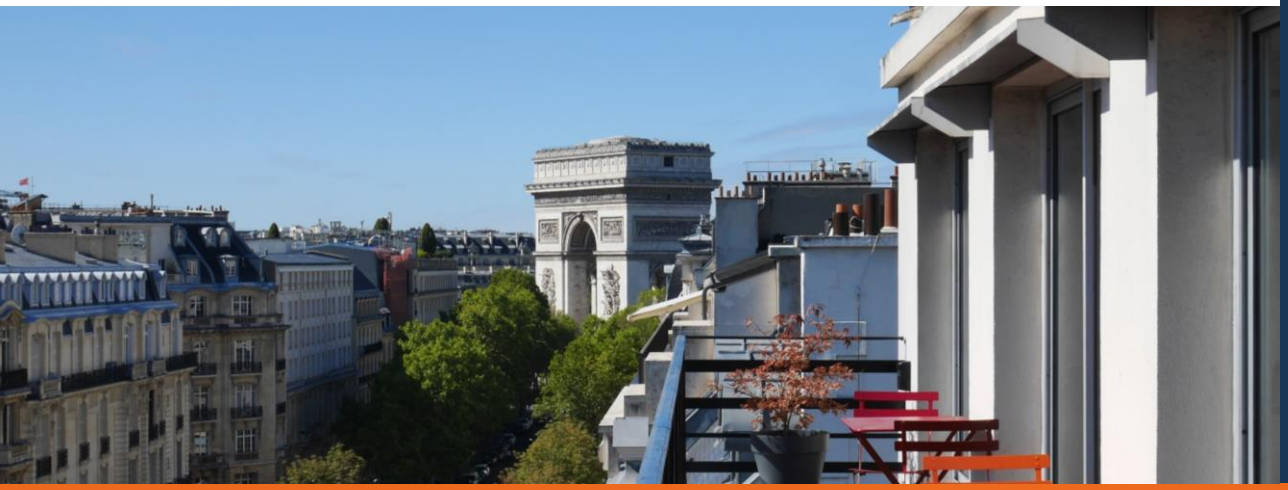
Julien Darlay

EURO Practitioners' Forum

hexaly

Software company specialized in Mathematical Optimization, Operations Research, and Decision Science

- Powerful optimization solver & platform used by Amazon, FedEx, Starbucks, …
- Turnkey, custom optimization and planning applications for Air Liquide, Toyota, …



- The fastest and most scalable solver for Routing, Scheduling, Packing, and more
- 20 years of experience
- 200 clients, 400 applications, and 20,000 users in 25 countries
- Offices in Brooklyn, NY, and Paris, France

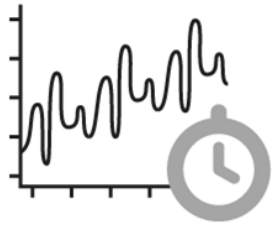
Context

Business model

Ads are served before or during a video content

Advertisers can buy a given number of impressions (=1 ad) for the next weeks

- ➔ Need a good estimate of this quantity
- ➔ Time series forecast
- ➔ Can be achieved with standard Data science / ML / statistics techniques



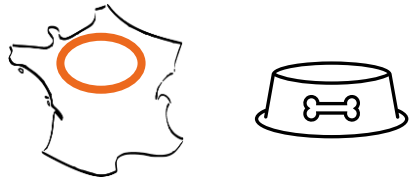
Context

Targeting

Advertisers are interested in a specific part of the population

- Socio demo characteristics
- Geolocation
- Buyers of product

18 - 25



Or any combination of categories: « Women between 18 and 25 who are pet owners and live in Paris or Marseille».

→ Need to compute the number of impressions for these requests

- Ten million impressions per week
- Millions of users with data
- Around 10000 user features (pet owners, men, …)
- Around 100ms to provide a good estimate

→ Need fast algorithms

Count distinct problem

Exact algorithm

Given a multiset S, estimate the number of distinct elements

- Eg.: S ={a, a, b, c, b, a } → distinct elements {a, b, c}
- Exact algorithm:
  - Iterate over all the elements and add them to a set
  - Complexity $O(Nlog(N))$ and $O(n)$ memory footprint

Business problem

- Consider all the impressions in the past months satisfying the user request (« pet owners between 18-25… »)
- Estimate the probability of satisfying the request
- Problem: we need to scan the whole database for each request

Count distinct problem

Approximate counting

Given a multiset S, estimate the number of distinct elements

- E.g.: S ={a, a, b, c, b, a } → 3 distinct elements {a, b, c}
- Hyperloglog algorithm [Flajolet et al. 2007]:
  - For each element $x \in S$
    - Compute $hash(x)$ → gives a 64 bits string, e.g. « $hash(a) = 0010011$ …»
    - Keep the number of leading zeros, e.g. 2 zeros
  - Keep the maximum number of leading zeros lz and return 2^{lz}
  - Complexity $O(N)$ with N the size of the multiset and $O(\log(\log(n))$ memory footprint
- Huge variance → can be reduced by averaging → theoretical guarantees
- Low memory footprint (~8bits for lz)
- Problem: still need to scan the whole database ☹

Approximate counting

Count distinct problem

Given 2 multisets S_1, S_2 estimate the number of distinct elements of $S_1 \cup S_2$

- Eg.: $S_1 =$ {a, a, b} and $S_2 =$ {c, b, a } → 3 distinct elements {a, b, c}
- Compute the number of leading zeros for $S_1 \rightarrow lz_1$, same for $S_2 \rightarrow lz_2$
- Use $lz \leftarrow \max(lz_1, lz_2)$ to estimate the number of distinct elements of $S_1 \cup S_2$

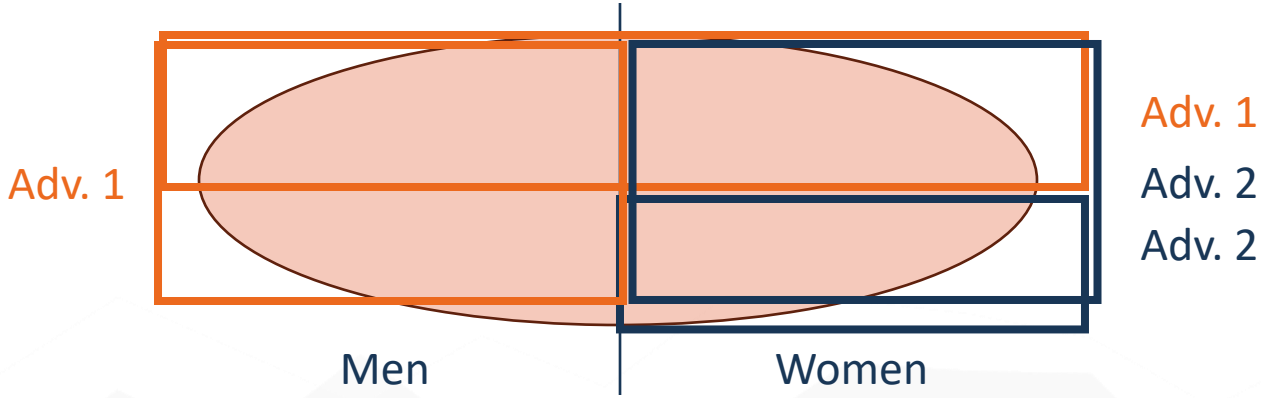
Business problem

- We can precompute lz for each feature independently ~10'000 values in a few hours
- We can compute $|A \cup B|$ in $O(1)$
- We can compute $|A \cap B| = |A \cup B| - |A| - |B|$
- « Women between 18 and 25 who are pet owners and live in Paris or Marseille» → CNF
- Practical results are better than the theoretical guarantees
- Can be extended to consider booked requests

Optimization problem

Consider a total of 1'000 impressions next week

- Advertiser 1 wants to book 500 impressions without any restrictions
- Advertiser 2 wants to book 500 impressions of "Women" (50% of the impression)
- Without optimization, Advertiser 1 will take 250 "Women" impressions



All the booked requests are reoptimized overnight using a greedy algorithm

Conclusion

Industrial project

- In production for 2 years now
- Used every day by multiple users

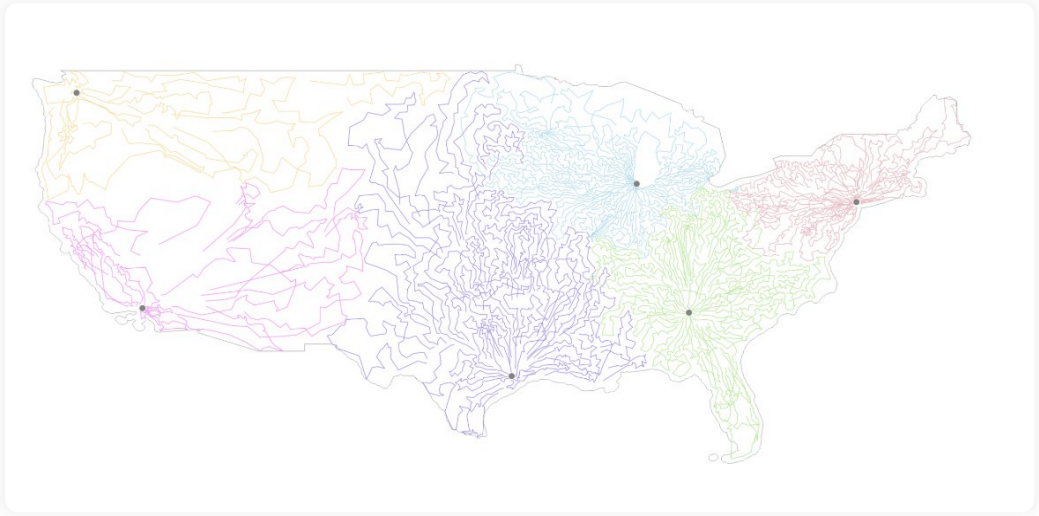
Project management

- Tight collaboration with the data science team
- Tight collaboration with business experts/end users

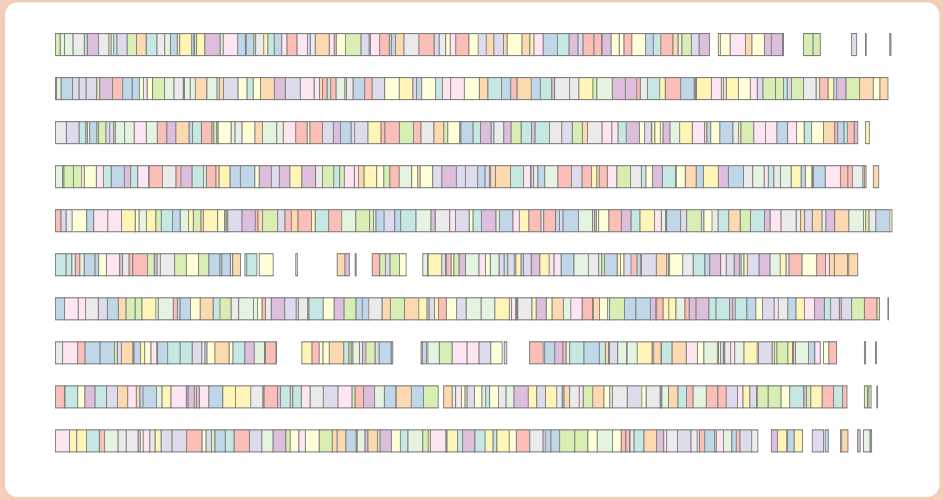
Optimization perspectives

- Use of approximate counting in optimization
- Faster approximate algorithms for `Max-k-coverage` and `Min-k-union`
- Compact models for these problems

ROUTING



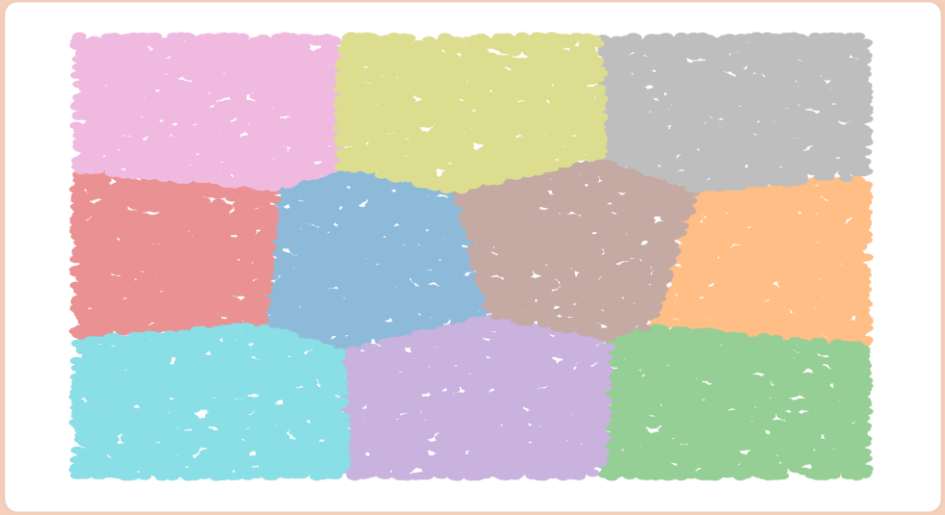
SCHEDULING



Questions?

PACKING

CLUSTERING



200 companies trust us