Approximate reconstruction of encrypted databases

Paul Grubbs, Marie-Sarah Lacharité, Brice Minaud, Kenny Paterson

Information Security Group

ROYAL HOLLOWAY UNIVERSITY OF LONDON

ESSA2, Bertinoro, 9th July 2018

Situation overview

General message from previous talk:

Don't use range queries with access pattern leakage!

Closer look:

- **KKNO16**: full reconstruction…
  - Assuming i.i.d. uniform queries.
  - $O(N^4 \log N)$ queries.
- **Kenny's talk**: full reconstruction…
  - Assuming density.
  - $O(N \log N)$ queries.

Approximate reconstruction

New goal: δ-approximate reconstruction.

Recover the values of records within δN.

LMP18 approximate attack but: only improvement in log factor, complicated analysis, requires density…

→ We would like to get best possible reconstruction with given queries. And handle large N's. And get rid of the density assumption, and i.i.d. queries.

Two new tools:

- **VC theory** (machine learning).
- **PQ-trees**.

Plan

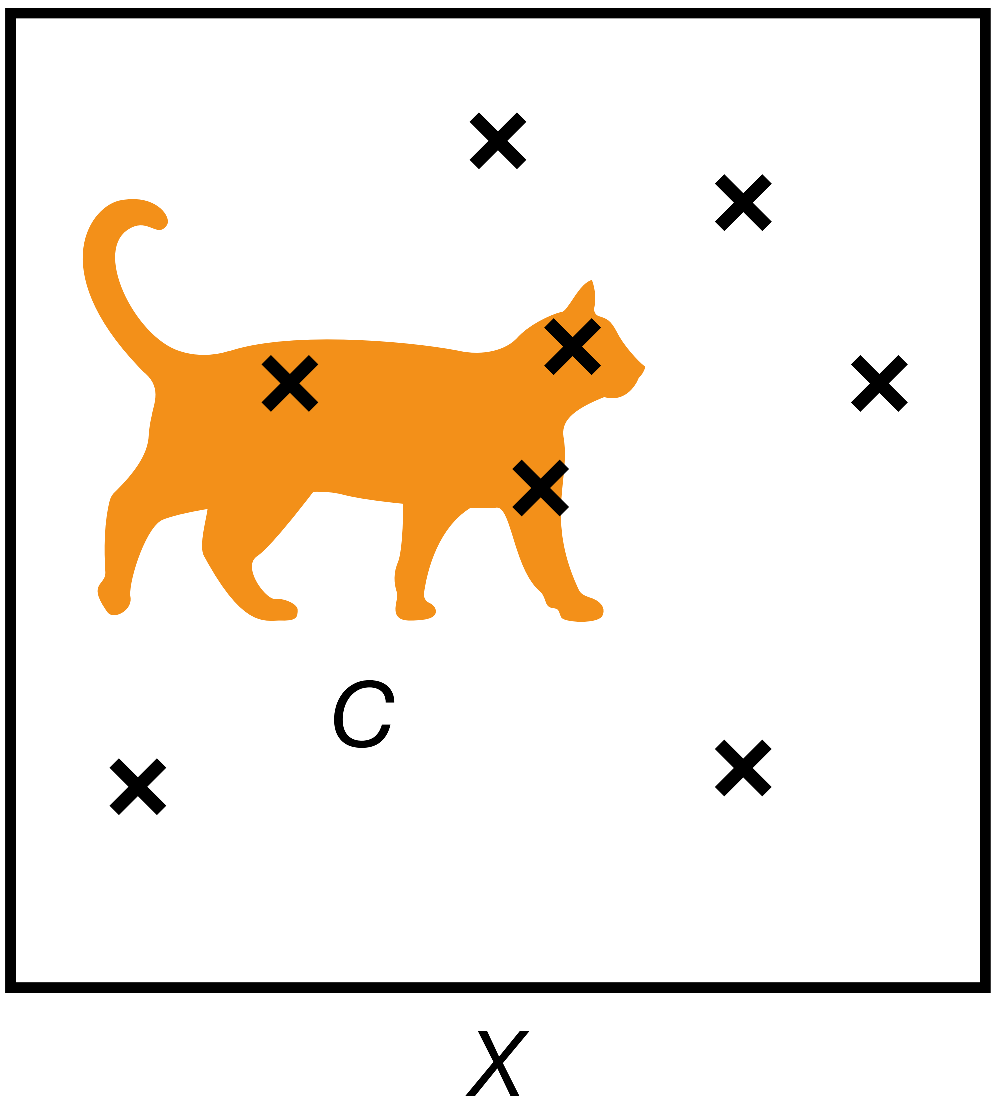
1. VC theory.
2. PQ trees.

VC theory

Warm-up

Set X with probability distribution D.

Let $C \subseteq X$. Call it a concept.

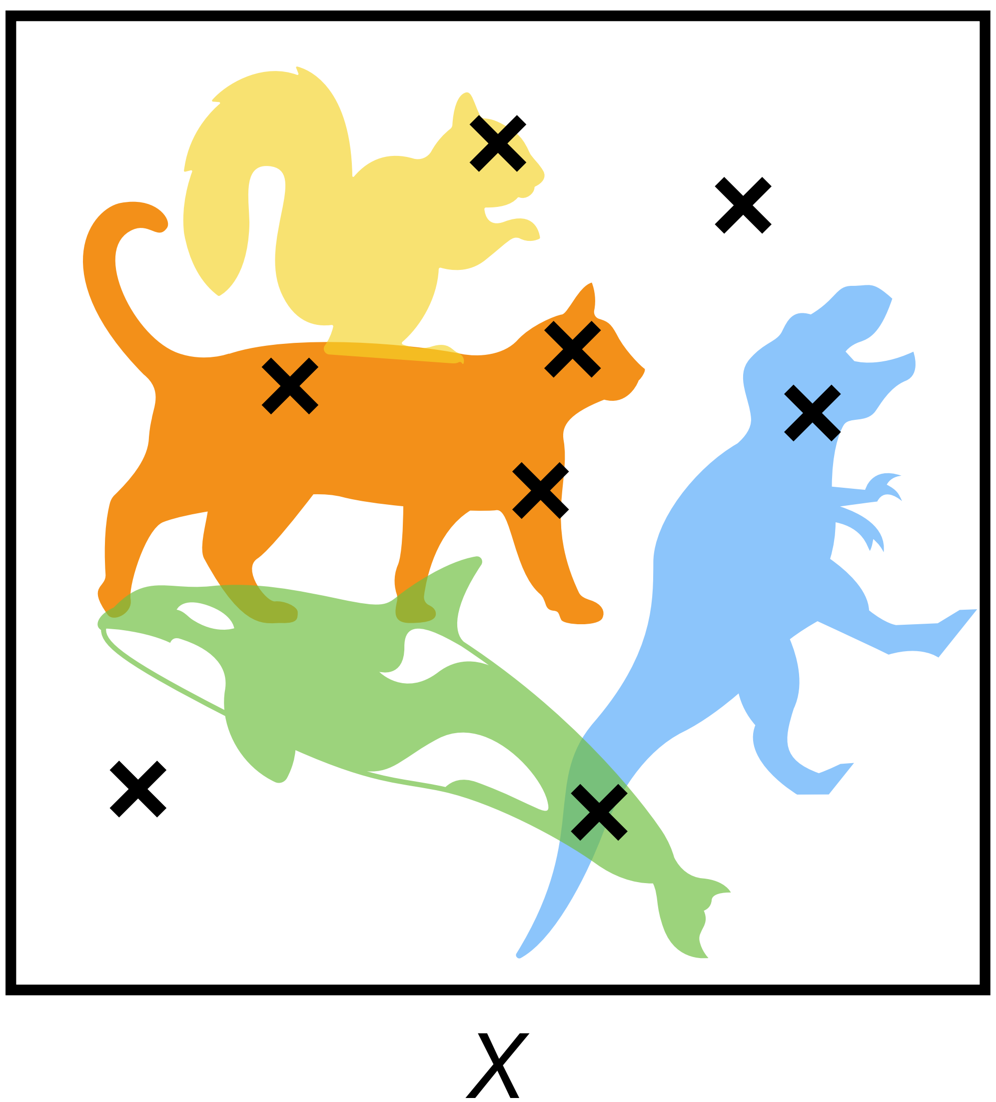


$$\Pr(C) \approx \frac{\#\text{points in } C}{\#\text{points total}}$$

Sample complexity:
to measure $\Pr(C)$ within δ,
you need $O(1/\delta^2)$ samples.

VC theory

Vapnik and Chervonenkis, 1971.
Now you have a set $\mathscr{C}$ of concepts.



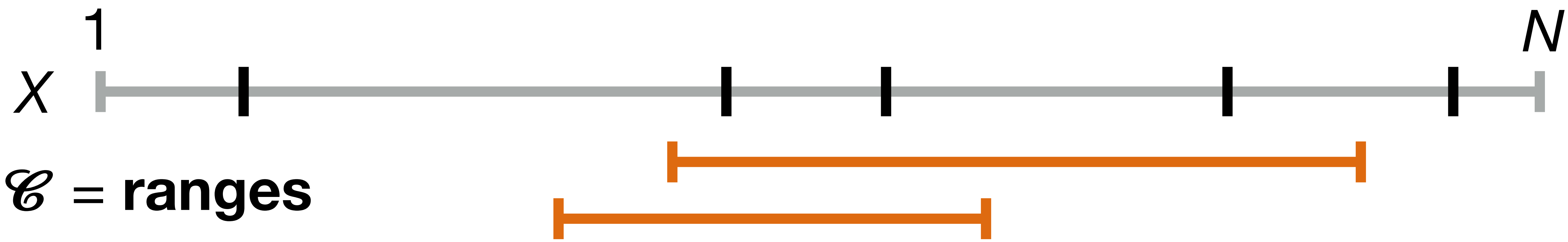
The set of samples drawn from X is an **ε-sample** iff for **all** C in $\mathscr{C}$:

$$\left| \Pr(C) - \frac{\#\text{points in } C}{\#\text{points total}} \right| \leq \epsilon$$

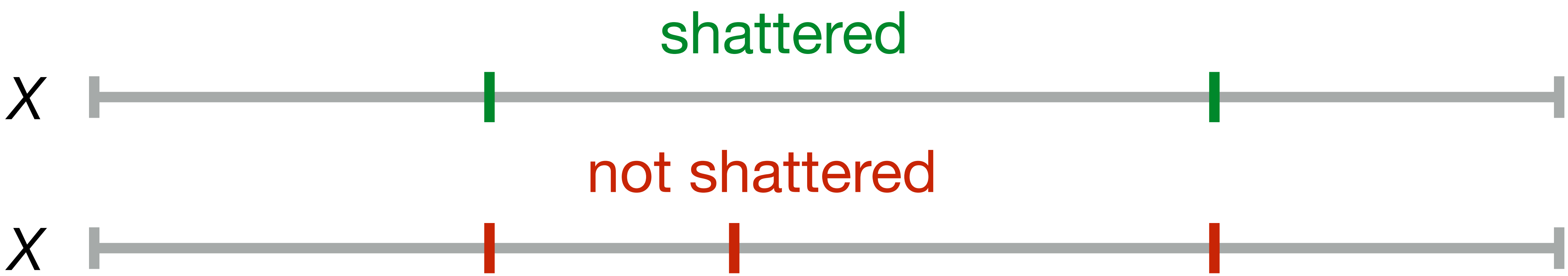
V & C 1971:

If $\mathscr{C}$ has **VC dimension** d, then the number of points to get an ε-sample whp is $O(d/\varepsilon^2 \log d/\varepsilon)$.

VC dimension



A set S of points in X is **shattered** by $\mathscr{C}$ iff every subset of S can be written in the form $C \cap S$ for some C in $\mathscr{C}$.



The **VC dimension** of $\mathscr{C}$ is the largest cardinality d such that every subset of X of size d is shattered.

e.g. for ranges the VC dimension is 2.

Two main results: ε-samples and ε-nets

The set of samples drawn from X is an **ε-sample** iff for **all** C in $\mathscr{C}$:

$$\left|\Pr(C) - \frac{\#\text{points in } C}{\#\text{points total}}\right| \leq \epsilon$$

→ If d is the VC dim, number of points to get an ε-sample whp is:

$$O\left(\frac{d}{\epsilon^2}\log\frac{d}{\epsilon}\right)$$

The set of samples drawn from X is an **ε-net** iff for **all** C in $\mathscr{C}$:

$$\Pr(C) \geq \epsilon \Rightarrow C \text{ contains a sample}$$

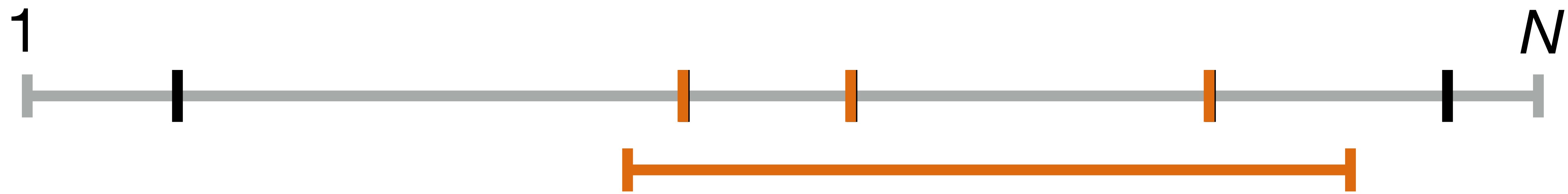
→ If d is the VC dim, number of points to get an ε-net whp is:

$$O\left(\frac{d}{\epsilon}\log\frac{d}{\epsilon}\right)$$

Example: learning range queries

Suppose we know the value of some records in the database (with uniformly random values).

+ we have access pattern leakage.



We want to **approximately learn** queries in the sense: for every query we want to know its endpoints within εN.

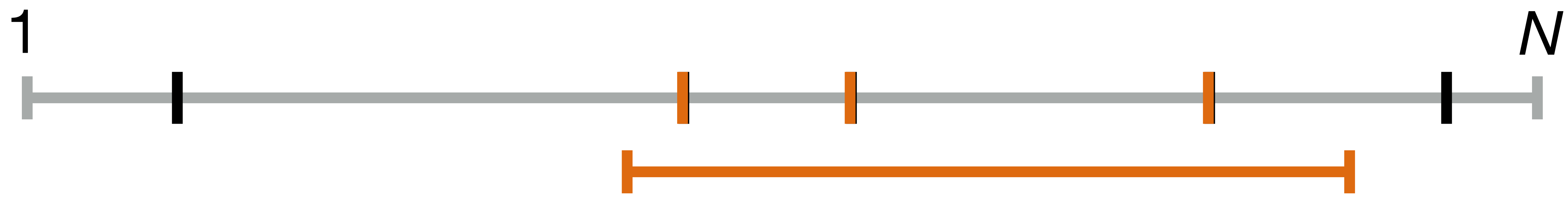
Q: How many known records do we need?

A: This is an ε-net.

X = values $[1,N]$ $\mathscr{C}$ = ranges

so we need $O(1/\varepsilon \log 1/\varepsilon)$ known samples.

Example continued



So this was an ε-net → we need $O(1/\varepsilon \log 1/\varepsilon)$ known samples.

Q: How about if we add complements? Multi-dimensional ranges? etc.

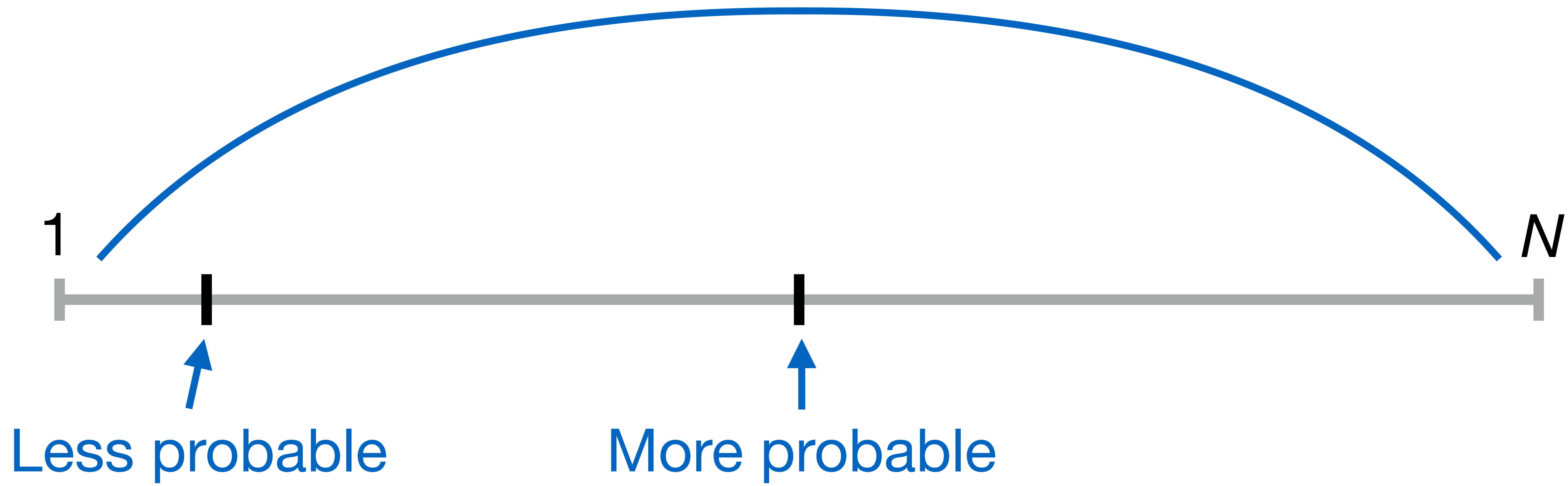
A: Actually we don't care. All these things have finite VC dim.

In fact this is *actually* **PAC learning**.

PAC = Probably Approximately Correct.

Database reconstuction

Basic KKNO16 attack variant

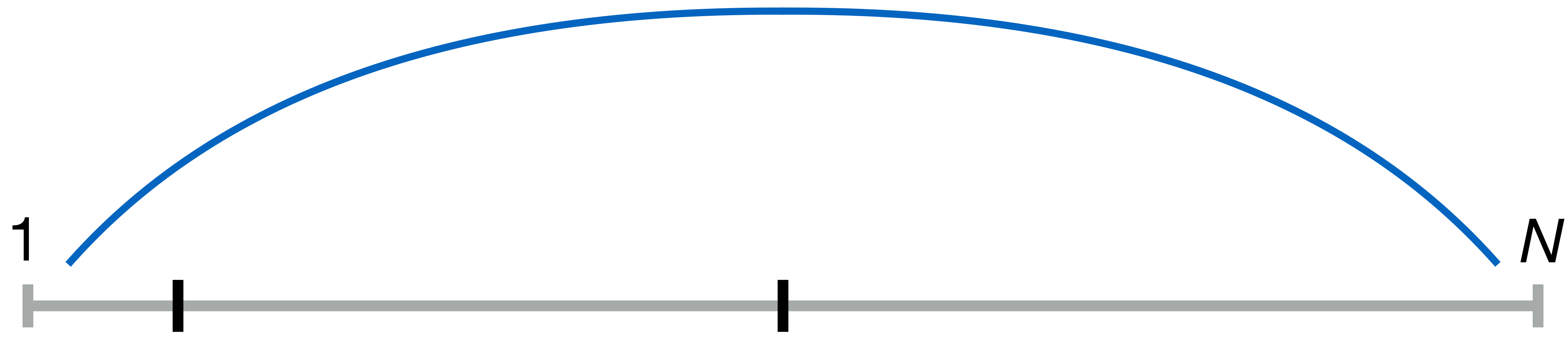


Assume **uniformly distributed** range queries.

Idea: count #times record is hit
→ estimate probability it's hit
→ deduce its value

Fact: to correctly deduce all values within δN you need to correctly estimate all probabilities within $\varepsilon = \delta^2$.

Basic KKNO16 attack variant



…so we need to estimate the probability of each value being hit, all within $\varepsilon = \delta^2$…

This is an ε-sample.

X = ranges $\qquad \mathscr{C} = \{\{\text{ranges} \ni x\} : x \in [1,N]\}$

so we need $O(1/\varepsilon^2 \log 1/\varepsilon)$ known samples.

Approximate KKNO attack

With **uniformly distributed** queries:

All values are in the database are recovered within δN after observing the access pattern of $O(1/\delta^4 \log 1/\delta)$ queries.

Remarks:

- KKNO16: $N^4 \log N$ ➞ Kenny's talk: $N \log N$ with density ➞ this: $O(1)$ for approximate reconstruction within 5%…
- Setting $\delta = 1/N$ recovers KKNO's attack.
- Lower bound of $\Omega(1/\delta^4)$.
- Direct application of VC theory.

Extensions of this approach

In fact $O(1/\delta^2 \log 1/\delta)$ queries suffice under very reasonable assumptions.

e.g. there exists record in DB with value within $[N/8, 3N/8]$.

Other query types:

- Prefix queries on strings, wildcard queries, etc.
- "Meta-theorem": all these have finite VC dim…
- This is WIP.

One limitation:

- VC theory gives bad constants.

It says something of general behavior. Need experiments.

Limitation of previous result

So far we are assuming **uniformly distributed** queries.

This is not just an assumption about adversarial knowledge.
This is an assumption that queries are **independent identically distributed** (i.i.d.).

This is quite unrealistic.

What can you learn without that hypothesis?

PQ trees

PQ trees

X: linearly ordered set. Order is unknown.

You are given a set S containing some some intervals in X.

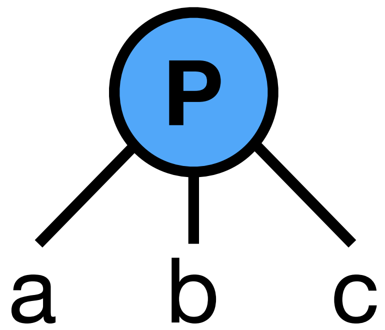
A PQ tree is a compact (linear in $|X|$) representation of the set of all permutations of X that are compatible with S.

As new sets are added to S, the PQ tree can be updated in linear time.

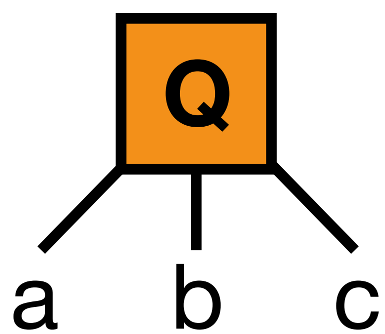
Was used in DR13, didn't target reconstruction.

PQ trees

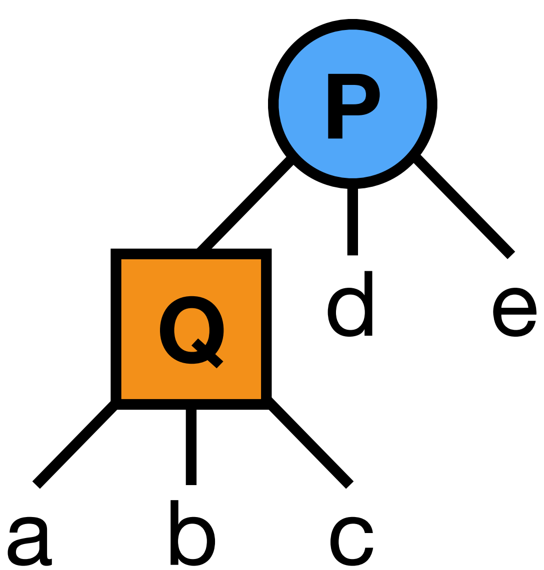
X = {a, b, c, d, e}



= any permutation of {a, b, c}.



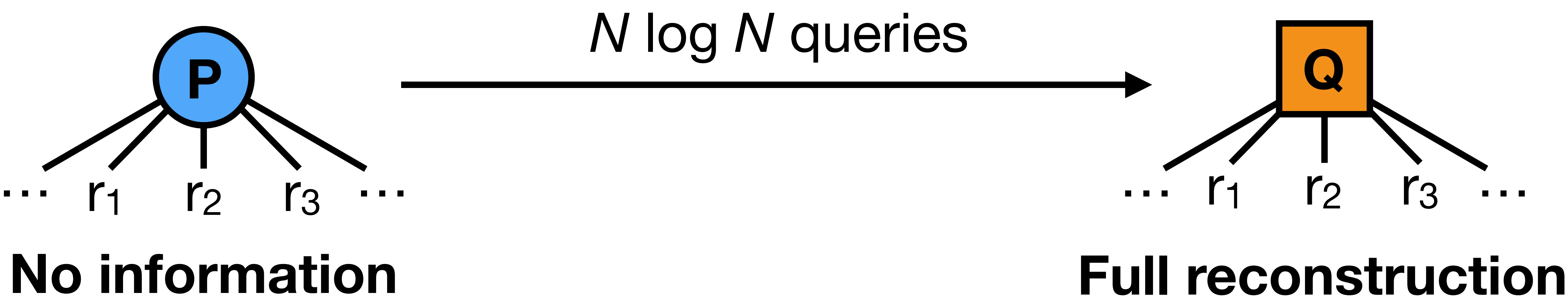
= ‘abc’ or ‘cba’.



= ‘abc’ or ‘cba’, with ‘d’ and ‘e’ permuted in any way on either side.

i.e. ‘abcde’, ‘abced’, ‘dabce’, ‘eabcd’, ‘deabc’, ‘edabc’, ‘cbade’ etc.

Database order reconstruction

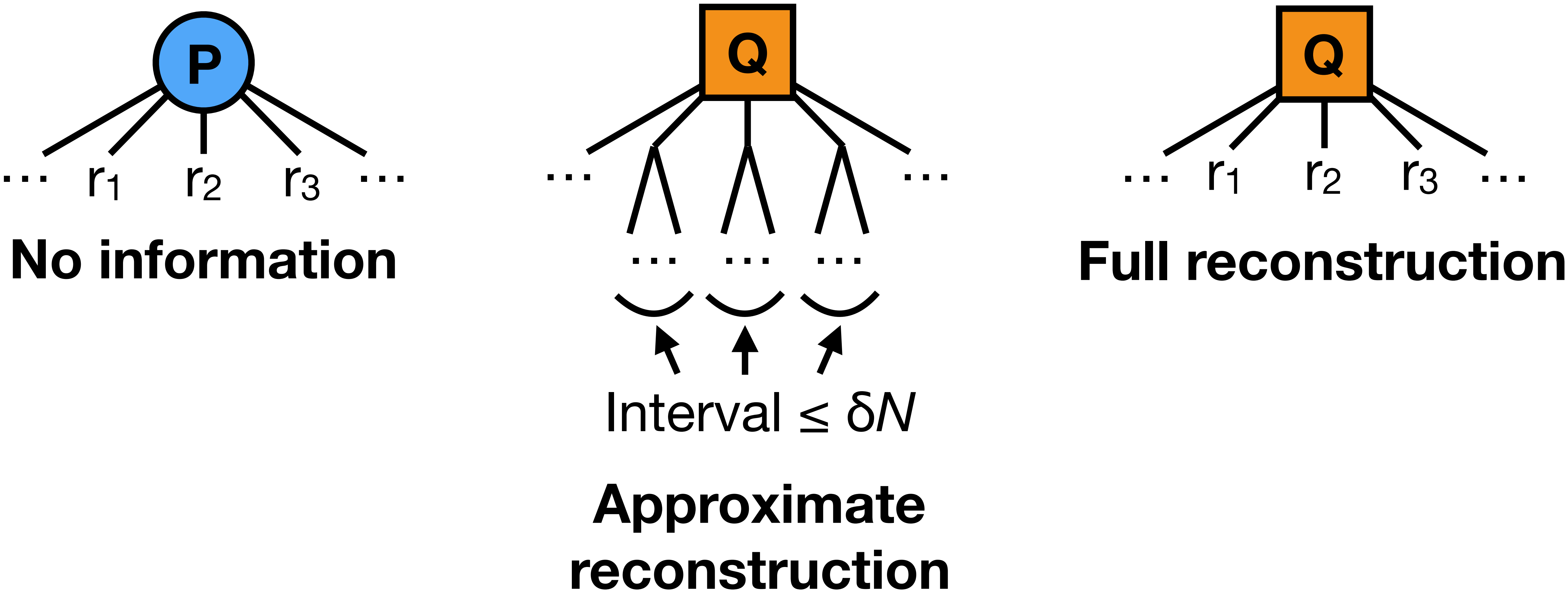


LMP18 (aka Kenny's talk) reinterpreted: you fully recover **order** information with $O(N \log N)$ queries.

Density not required.

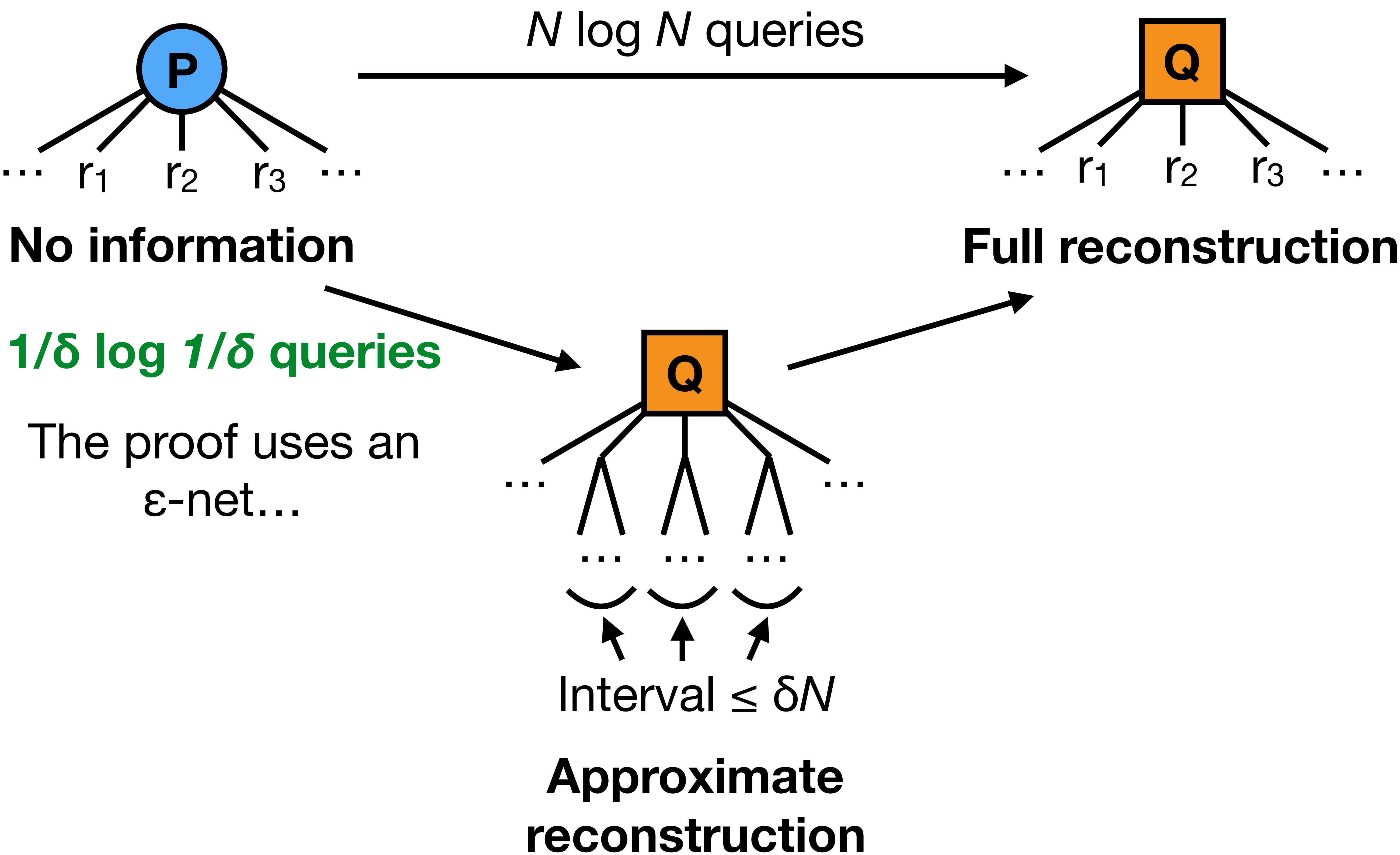
Density was only to convert from **order** to **values**.

Approximate order reconstruction



Approximate (order) reconstruction = full order reconstruction, except for values that are very close (less than δN apart).

Approximate order reconstruction



Converting from order to values

Known (approximation of) database value distribution → frequency matching.

Known (approximation of) query distribution, see previous attack.

Some known records → order allows to compare records to known values.

…

Some history

OPE/ORE were developed to allow range queries. Leak order by design. Led to devastating **leakage-abuse attacks** GSB+17, DDC16.

Second-generation schemes eschew ORE to enable range queries without leaking order.

We just saw access pattern leaks order… So if you leak access pattern it's back to square one!

(Difference: OPE/ORE attacks only required a snapshot adversary, now we need access pattern leakage.)

Features of the approximate order attack

It is **fully** general:

- Does not rely on i.i.d. queries.
- No density assumption.
- No dependency on N (for approximate order).

Also…

- Only $O(1/\delta \log 1/\delta)$ queries!
- Setting $\delta = 1/N$ recovers LMP18. Without requiring density.
- Not "all or nothing": precision improves with #queries.

Conclusion

Introduced approximate reconstruction.

Leads to very powerful attacks. Approximate order attack is very efficient with truly minimal assumption. Clarifies the setting.

Two techniques prove very potent in this setting:

- VC theory.
- PQ trees.

VC theory extends to other query classes (under investigation).