# “Classy” sample correctors[1]


Ronitt Rubinfeld
MIT and Tel Aviv University

joint work with Clement Canonne (Columbia) and Themis Gouleakis (MIT)

[1]thanks to Clement and G for inspiring this classy title

# Distributions on BIG domains

- Given samples of a distribution, need to know, e.g.,
  - entropy
  - number of distinct elements
  - “shape” (monotone, bimodal,…)
  - closeness to uniform, Gaussian, Zipfian…
  - learn parameters

- Considered in statistics, information theory, machine learning, databases, algorithms, physics, biology,…

# Key Question

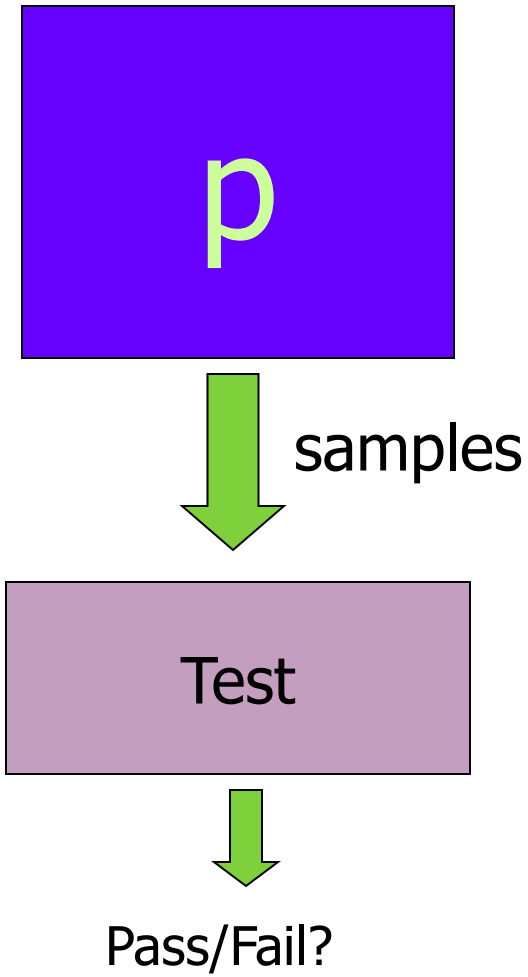
- How many samples do you need in terms of domain size?
  - Do you need to estimate the probabilities of each domain item?

  -- OR --

  - Can sample complexity be *sublinear* in size of the domain?

Rules out standard statistical techniques

# Our usual model:




- $p$ is arbitrary black-box distribution over $[n]$, generates iid samples.
- $p_i$ = Prob[ $p$ outputs $i$ ]
- Sample complexity in terms of $n$?

# Great Progress!

- Some optimal bounds:
  - Additive estimates of entropy, support size, closeness of two distributions: *n/log n* [Raskhodnikova Ron Shpilka Smith 2007][Valiant Valiant 2011]
  - Two distributions - the same or far (in L1 distance)? $n^{\frac{1}{2}}, n^{\frac{2}{3}}$ [Goldreich Ron][Batu Fortnow R. Smith White 2000] [Valiant 2008]
  - $\gamma$-multiplicative estimate of entropy: $n^{1/\gamma 2}$ [Batu Dasgupta Kumar R. 2005] [Raskhodnikova Ron Shpilka Smith 2007] [Valiant 2008]
  - And much much more!!

# So now what do you do?

You tested your distribution, and it's pretty much ok,

BUT

What if your samples aren't *quite* right?

# What are the traffic patterns?

Some sensors lost power, others went  crazy!

# Astronomical data

A meteor shower confused some of the measurements

# Teen drug addiction recovery rates

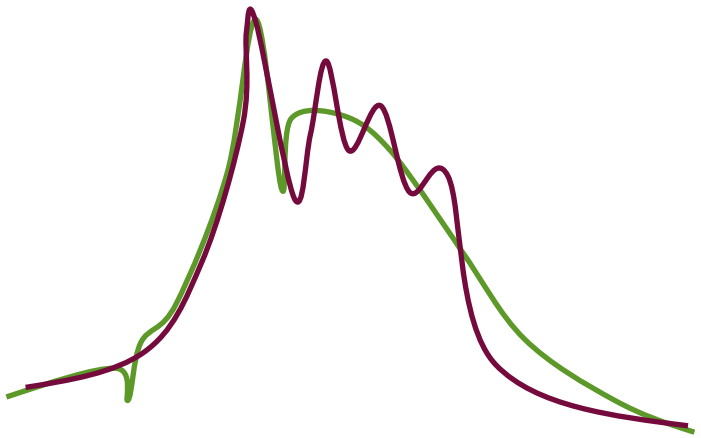

Never received data from three of the community centers!

# Whooping cranes

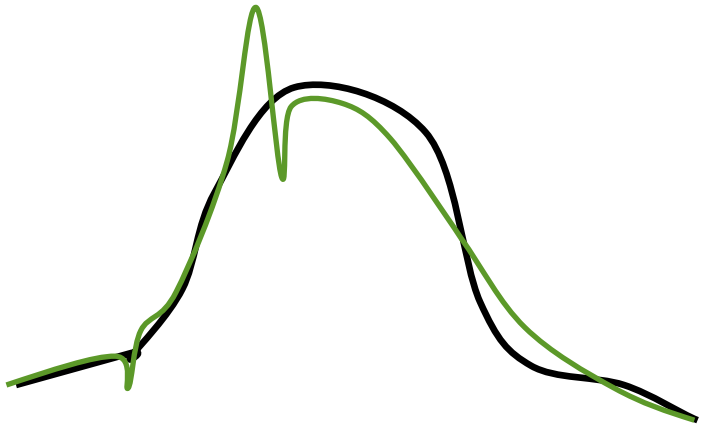
Correction of location errors for presence-only species distribution models

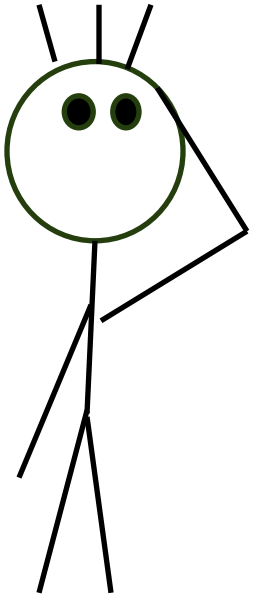
[Hefley, Baasch, Tyre, Blankenship 2013]

# What is correct?



# What is correct?



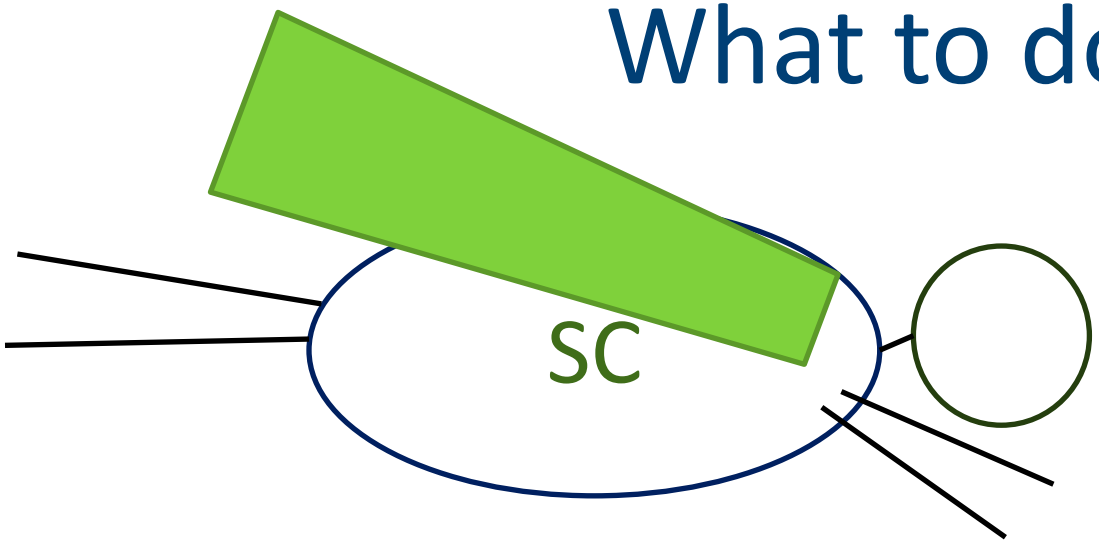


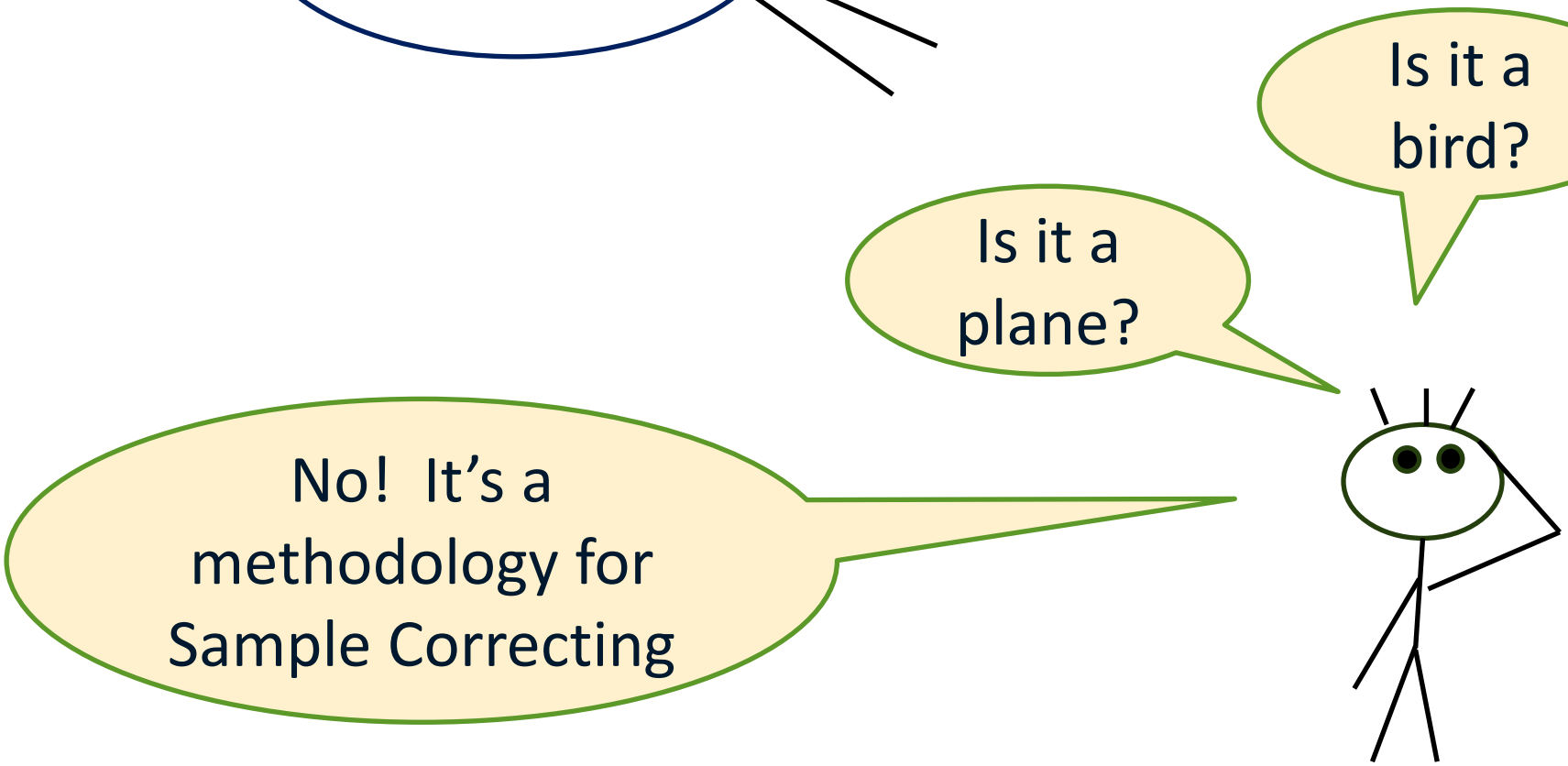
# What to do?

- Outlier detection/removal
- Imputation
- Missingness
- …

What if you don’t know that the distribution is supposed to be normal, Gaussian, …?

# What to do?


SC


Is it a bird?
Is it a plane?
No! It's a methodology for Sample Correcting

# What is correct?

Sample corrector assumes that original distribution in class *P*

(e.g., *P* is monotone, Lipshitz, *k*-modal, *k*-histogram distributions)

# Classy Sample Correctors

- Given: Samples of distribution $q$ assumed to be $\epsilon$-close to class $P$

- Output: Samples of some $q'$ such that
  - $q'$ is $\epsilon'$-close to distribution $q$
  - $q'$ in $P$

1. Sample complexity per output sample of $q'$?
2. Randomness complexity per output sample of $q'$?

# An observation

Agnostic learner  Sample corrector

Corollaries: Sample correctors for
- monotone distributions
- histogram distributions under promises (e.g., distribution is MHR or monotone)

# The big open question:

When can sample correctors be *more* efficient than agnostic learners?

- Some answers for monotone distributions:
  - Error is REALLY small
  - Have access to powerful queries
  - Missing data errors
  - Unfortunately, not likely in general case (constant arbitrary error, no extra queries)

# Learning monotone distributions

Learning monotone distributions requires $\theta(\log n)$ samples

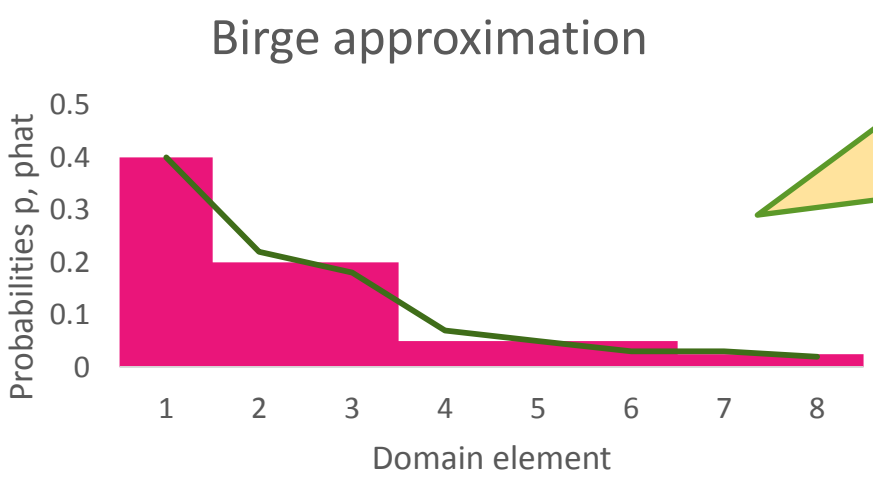
[Birge][Daskalakis Diakonikolas Servedio]

# Birge Buckets

Partition of domain into buckets (segments) of size $(1+\epsilon)^i$
($O(\log n)$ buckets total)

For distribution $p$, let $\hat{p}$ be such that uniform on each bucket, but same marginal in each bucket

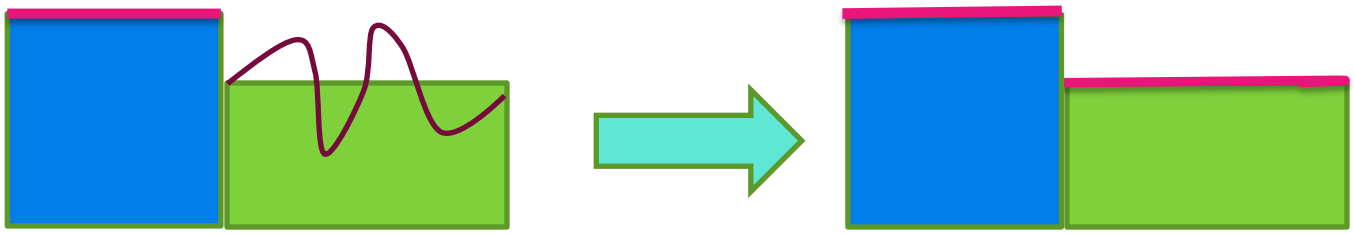
Then $\left|\left|p-\hat{p}\right|\right| \leq \epsilon$




Enough to learn the marginals of each bucket

# A very special kind of error

Suppose ALL error located internally to Birge Buckets



Then, easy to correct to $\hat{p}$:

1. Pick sample *x* from *p*
2. Output y chosen UNIFORMLY from *x*'s Birge Bucket

"Birge Bucket Correction"

# Learning monotone distributions

Thm: Exists Sample Corrector which given *p* which is $\left(\frac{1}{\log^2 n}\right)$ –close to monotone, uses *O(1)* samples of *p* per output sample.

OBLIVIOUS CORRECTION!!

Proof Idea:

Mix Birge Bucket correction with slightly decreasing distribution (flat on buckets with some space between buckets)

# A recent lower bound [P. Valiant]

Sample correctors for $\Omega(1)$-close to monotone distributions require $\Omega(\log n)$ samples

What do we do now?

# What about stronger queries?

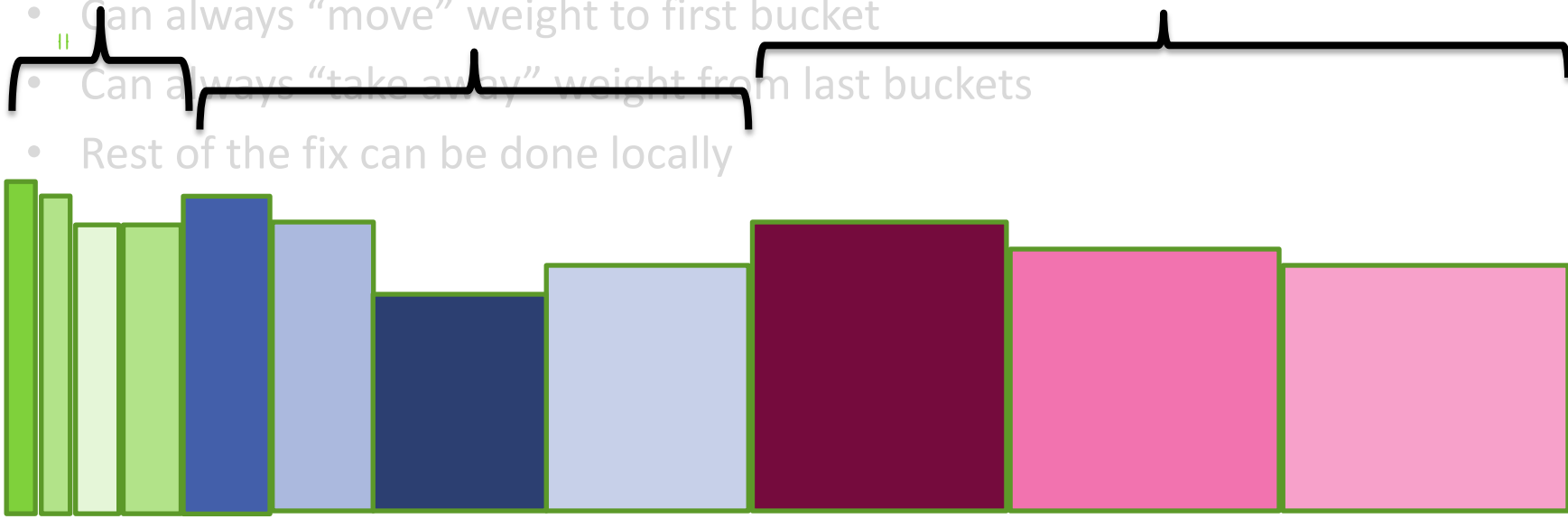
What if have lots and lots of sorted samples?

Easy to implement both samples, and queries to cumulative distribution function (cdf)!

Thm: Exists Sample Corrector such that given *p* which is $\epsilon$ −close to monotone, uses $O((\log(n))^{1/2})$ queries to *p* per output sample.

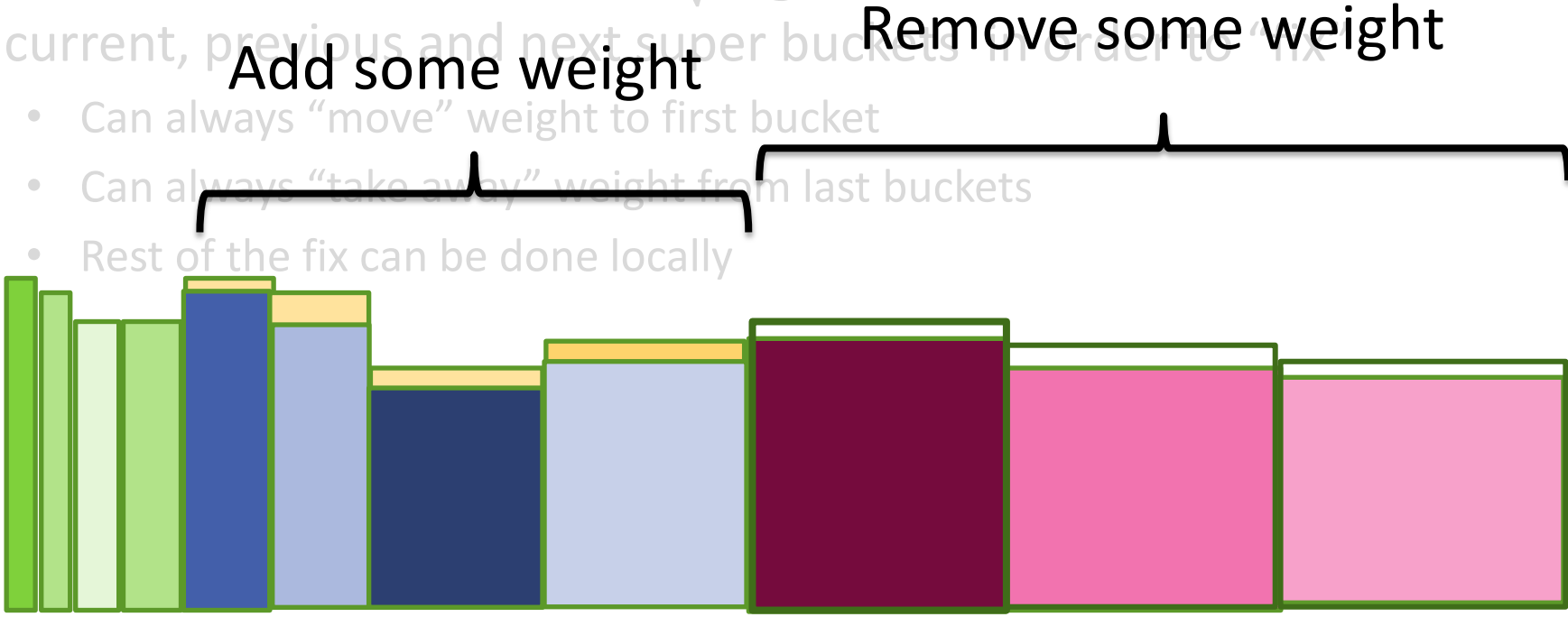
# Fixing with CDF queries

- Each *super bucket* is $\sqrt{\log n}$ consecutive Birge buckets
- Query conditional distribution of superbuckets and reweight if needed
- Within super buckets, use $O(\sqrt{\log n})$ queries to all buckets in current, previous and next super buckets in order to "fix"
  - Can always "move" weight to first bucket
  - Can always "take away" weight from last buckets
  - Rest of the fix can be done locally



# Fixing with CDF queries

- Each *super bucket* is $\sqrt{\log n}$ consecutive Birge buckets
- Query conditional distribution of superbuckets and reweight if needed (decide how using LP)
- Within super buckets, use $O(\sqrt{\log n})$ queries to all buckets in current, previous and next super buckets in order to "fix"
  - Can always "move" weight to first bucket
  - Can always "take away" weight from last buckets
  - Rest of the fix can be done locally



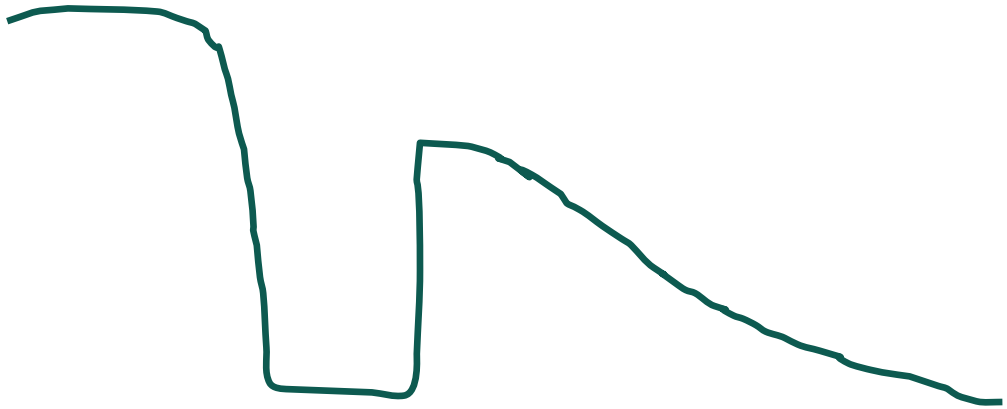
# Fixing with CDF queries

- Each *super bucket* is $\sqrt{\log n}$ consecutive Birge buckets
- Query conditional distribution of superbuckets and reweight if needed
- Within super buckets, use O($\sqrt{\log n}$) queries to all buckets in current, previous and next super buckets in order to "fix"
  - Can always "move" weight to first bucket, "take away" weight from last buckets
  - Rest of the fix must be done *quickly* and *on the fly*…
    - After reweighting above, average weights $a_i$ of a superbucket are monotone
    - Ensure that new corrections don't violate monotonicity with the $a_i$'s

# Special error classes

- Missing data errors – *p* is a member of *P* with a segment of the domain removed
  - E.g. one sensor failure in traffic data

*More efficient sample correctors via learning missing part*



# Sample correctors provide more powerful learners and testers:

- Sample Corrector + learner → agnostic learner
- Sample Corrector + distance approximator + tester → tolerant tester
  - Gives weakly tolerant monotonicity tester

# Randomness Scarcity

- Can we correct using little randomness of our own?
  - Generalization of Von Neumann corrector of biased coin
  - Compare to extractors (not the same)
  - For monotone distributions, YES!

# What next for correction?

When is correction easier than learning?

Thank you