Learning **S**ums of **I**ndependent **C**ommonly **S**upported **I**nteger **R**andom **V**ariables (**SICSIRV**s)

Anindya De
Northwestern University

Based on joint work with

Philip Long
Google Research

Rocco Servedio
Columbia University

What this talk is about:

Learning certain types of

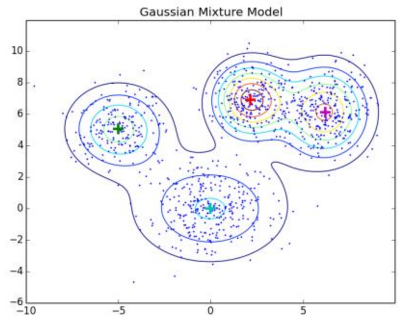
discrete probability distributions

from random examples.

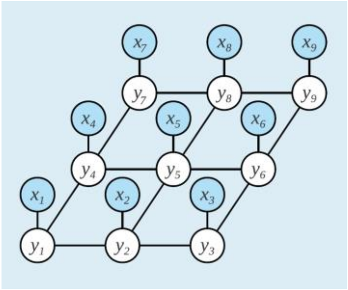
Outline of the talk

- What we mean by learning a discrete distribution
- Related distributions, relevant prior work
  - The particular kinds of distributions we consider: **SICSIRVs**
- Our results: Algorithms and lower bounds
- Some ideas that underlie the results

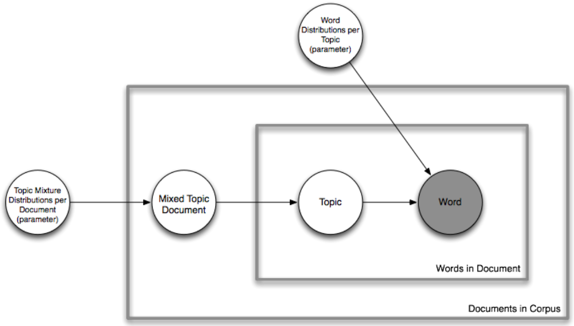
Learning probability distributions



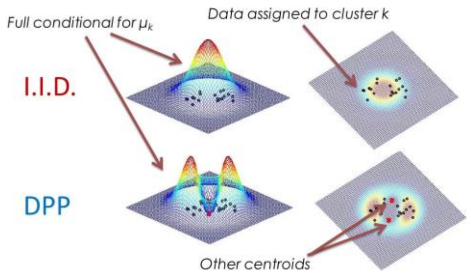
Gaussian mixture models



Markov Random Fields



Latent Dirichlet Allocation (LDA)



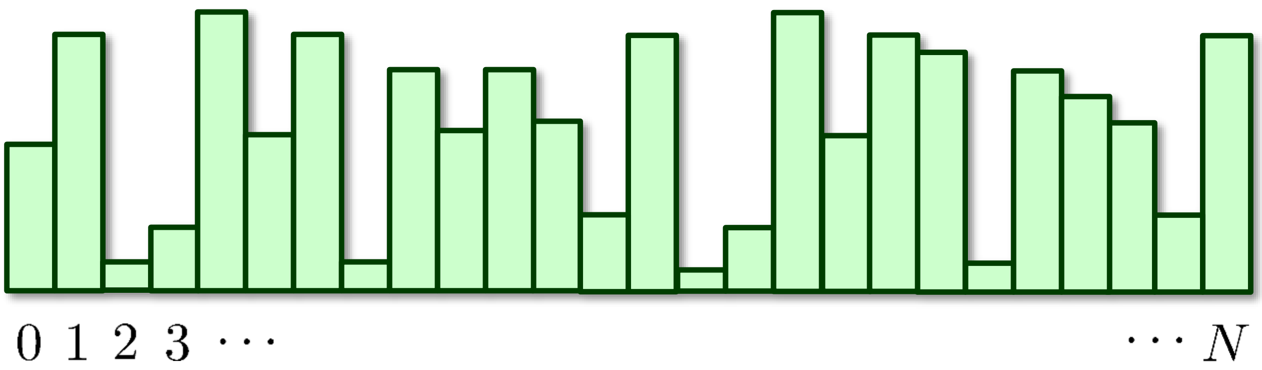
Determinantal point process (DPP)

This talk

- Complexity theoretic take on the problem.
- Distributions generated by computationally simple processes.
- Interesting phenomenon on sample complexity emerges.

Learnability of discrete distributions

- Discrete distributions: for us, distributions over $\mathbb{Z}$.

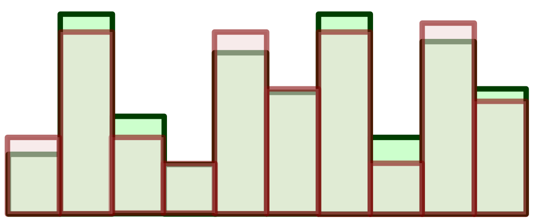


- A learning problem is defined by a class $\mathcal{C}$ of distributions.

  An instance of the learning problem corresponds to an unknown target distribution $\mathcal{D} \in \mathcal{C}$.

The learning game

- Learner gets i. i. d. draws from distribution $\mathcal{D}$.
- Aim: with probability 9/10, the learner produces a hypothesis $\mathcal{D}'$ such that $\|\mathcal{D} - \mathcal{D}'\|_1 \leq 1/10$.



Equivalently,
statistical distance or
total variation distance

Natural question:

What classes of distributions can be learned **efficiently?**

(fast running time, using few examples)

Getting our feet wet

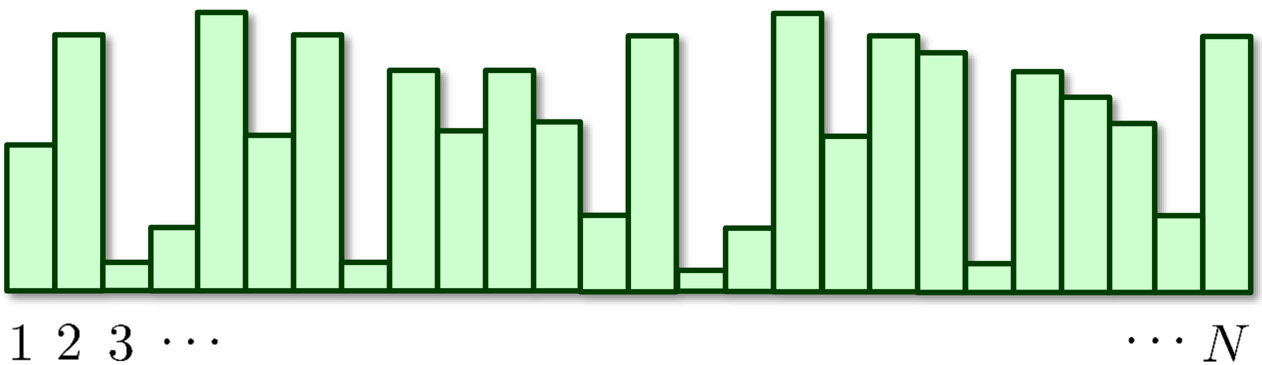
The absolute most basic case: $\mathcal{C}$ = all Bernoulli distributions (distributions over {0,1})

- Equivalent to learning unknown bias of a coin

- $\Theta(1/\epsilon^2)$ samples are sufficient (and essentially necessary) for learning to total variation distance ϵ

Another simple example

$\mathcal{C}$ = all distributions supported on {1,2,…,N}



– Well known that $\Theta(N/\epsilon^2)$ examples are sufficient (and again, essentially necessary)

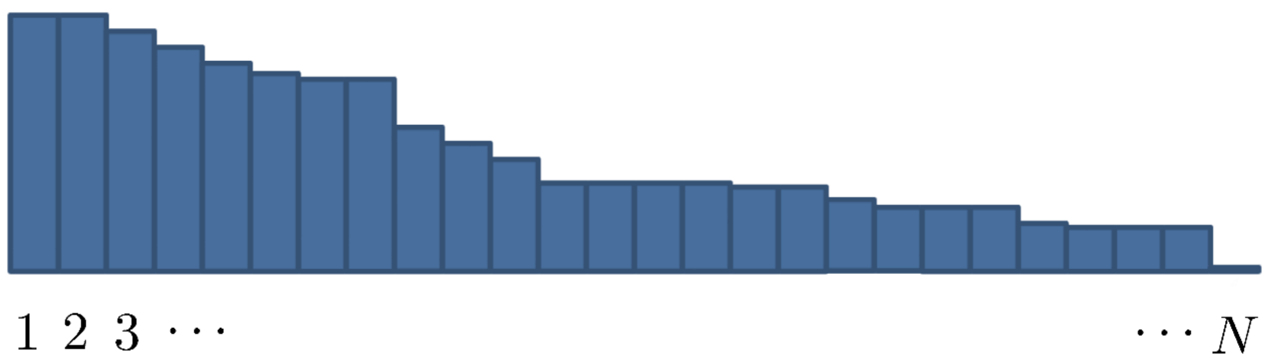
Brute force methods

- Both these are examples of brute force methods.

- Algorithm just outputs the empirical estimate – if point x appears in the sample t_x fraction of times, then the hypothesis $D'(x) = t_x$.

Last example

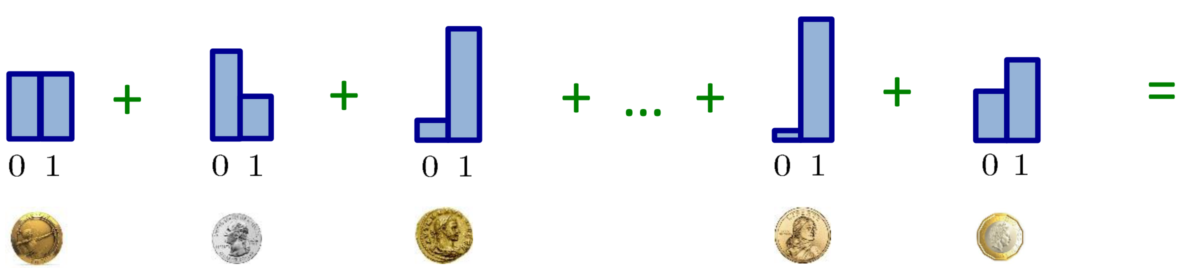
$\mathcal{C}$ = all *monotone non-increasing* distributions supported on {1,2,…,N}



– $\Theta(\log(N)/\varepsilon^3)$ samples necessary and sufficient for learning [Birge88]

So, what is a SICSIRV?

- We'll talk about it later… Let's begin with a special case.
- Consider the family of *Poisson Binomial distributions:* Sums of N independent Bernoulli random variables.
- I.e., each sample distributed as $\mathbf{X}_1 + \cdots + \mathbf{X}_N$ where $\mathbf{X}_1, \ldots, \mathbf{X}_N$ are independent $\{0,1\}$ r.v.s



PBD

Example: Newspaper circulation


THE GROWTH STORY KNOWHOW PLUS HOLY DIPLOMACY FOREIGN

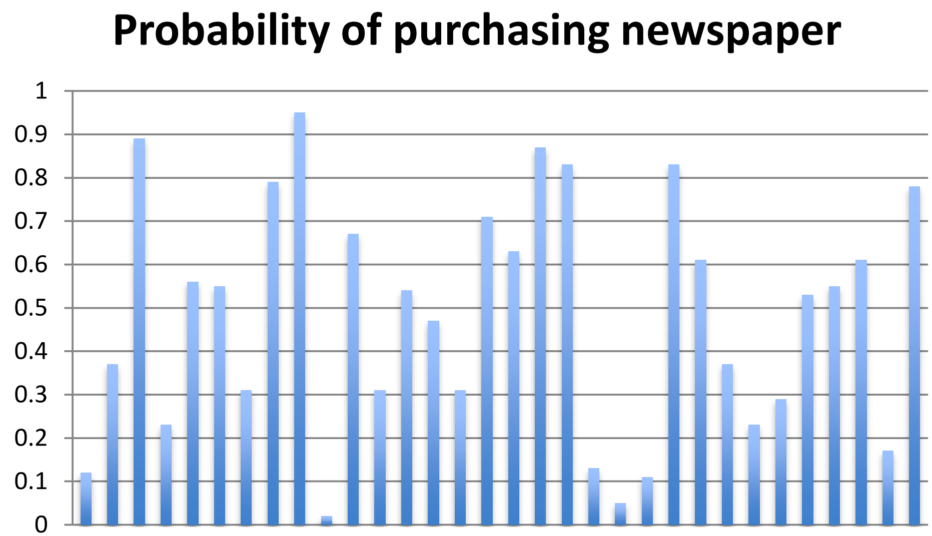
The Telegraph

Photo ban has blanket reach

That mother of all extinctions

Millions of years ago, an Indian valley in the Himalaya felt repercussions of events that wiped out most species

Hello, Cinderella. Sorry, I am Grace



Probability of person i purchasing newspaper = p_i (bias of random variable $\mathbf{X}_i$).

Total circulation random variable $\mathbf{X_1 + X_2 + X_3 + ... + X_n}$

Learning Poisson Binomial Distributions

Theorem: [DDS12] The time and sample complexity of learning Poisson Binomial distributions is poly$(1/\epsilon)$.

This complexity is **independent of N!**

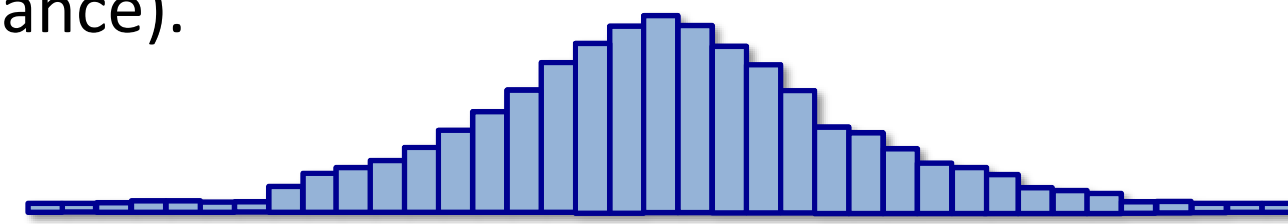
Intuition: Either

(i) The target distribution has large variance, i.e. variance $\geq$ poly$(1/\epsilon)$.

(ii) Or target distribution has small variance $\leq$ poly$(1/\epsilon)$.

Case Analysis

- Large variance (non-degenerate case): If the variance is at least poly$(1/\epsilon)$, then the distribution is $O(\epsilon)$ close to a discretized Gaussian (with the population mean and variance).



Discretized Gaussian

- Small variance (degenerate case): If the variance is at most poly$(1/\epsilon)$, then the effective support is poly$(1/\epsilon)$.

Learning PBDs, cont

- Large variance (non-degenerate case): Reduces to learning a (basically) Gaussian distribution.
  Learning both the mean and variance to error ϵ takes poly$(1/\epsilon)$ samples.

- Small variance (degenerate case): The size of the effective support is poly$(1/\epsilon)$. Can be learned by brute force in time poly$(1/\epsilon)$.

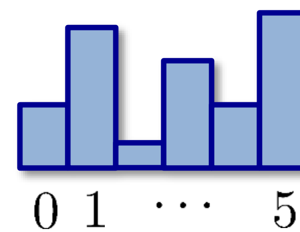
Hypothesis testing

(Informally): If there is a distribution D (i. i. d. sample access) and candidate distributions D_1, D_2, … , D_k such that at least one is close to D, then you can figure out which one using O(log k) sample overhead.

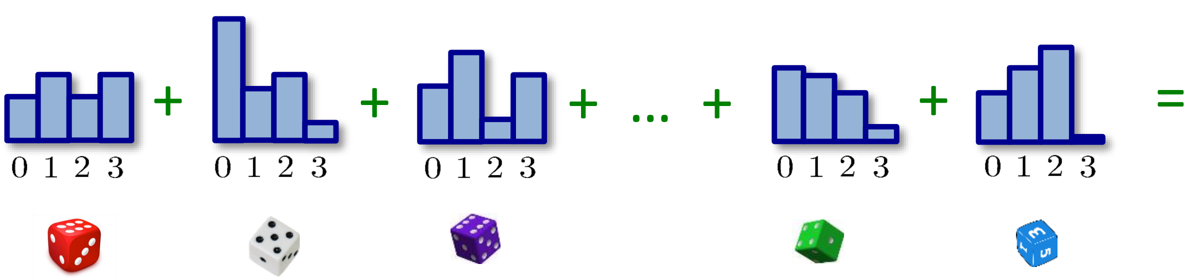
The next step: k-SIIRVs

k-IRV: Integer-valued Random Variable supported on $\{0, 1, \ldots, k-1\}$

6-IRV

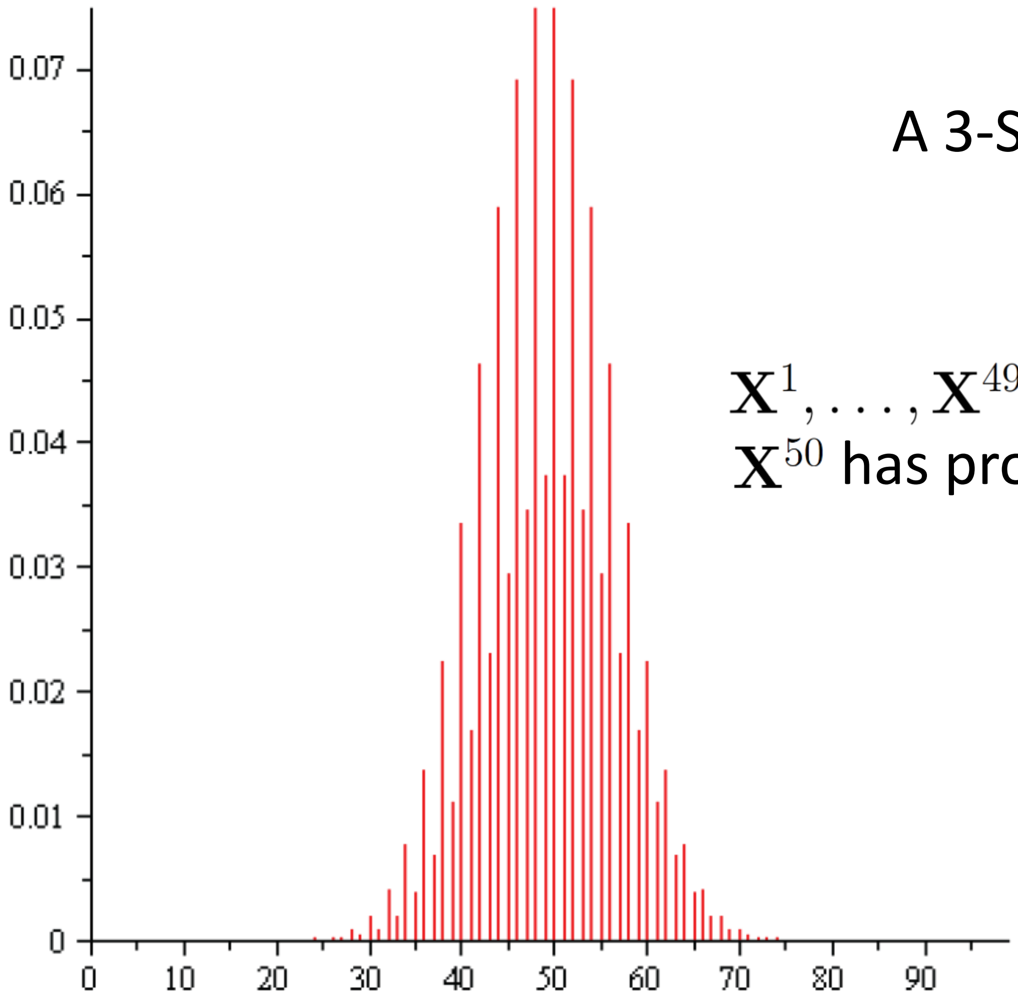


k-SIIRV: Sum of N Independent (*not necessarily identical*) *k*-IRVs



4-SIIRV

Example



A 3-SIIRV with N=50.

$\mathbf{X}^1, \ldots, \mathbf{X}^{49}$ each uniform over {0,2}
$\mathbf{X}^{50}$ has probability 2/3 on 0,
1/3 on 1.

Learning algorithm for k-SIIRVs

Theorem: [DDOST13] Let $\mathcal{C}$ be the class of *k*-SIIRVs, *i.e.* all distributions

$$\mathbf{S} = \mathbf{X}_1 + \cdots + \mathbf{X}_N$$

where the $\mathbf{X}_i$'s are independent random variables each supported on $\{0, 1, \ldots, k-1\}$.
There is an algorithm that learns $\mathcal{C}$ with time and sample complexity $\mathrm{poly}(k, 1/\varepsilon)$, independent of N.

Heart of [DDOST13]:

A new structure theorem for *k*-SIIRVs:

"Every *k*-SIIRV is close to
sum of two simple independent random variables"

Structure Theorem. Let $\mathbf{S}$ be a *k*-SIIRV with

$$\mathrm{Var}[\mathbf{S}] \geq \mathrm{poly}(k/\varepsilon).$$

Then $\mathbf{S}$ is ε-close to $c\mathbf{Z} + \mathbf{Y}$, where

- $c \in \{1, \ldots, k-1\}$
- $\mathbf{Z} =$ discretized Gaussian
- $\mathbf{Y} = c$-IRV
- $\mathbf{Y}, \mathbf{Z}$ independent

$c\mathbf{Z}$: discretized Gaussian scaled by c

$\mathbf{Y}$: supported on $\{0, 1, \ldots, c-1\}$

It’s SICSIRV time

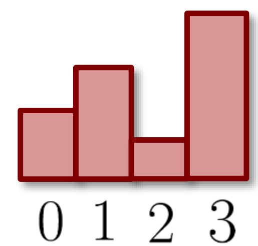
Here’s what a SICSIRV is:

Given non-negative integers $0 \leq a_1 < \cdots < a_k$,
a SICSIRV over $\{a_1, \ldots, a_k\}$ is a random variable

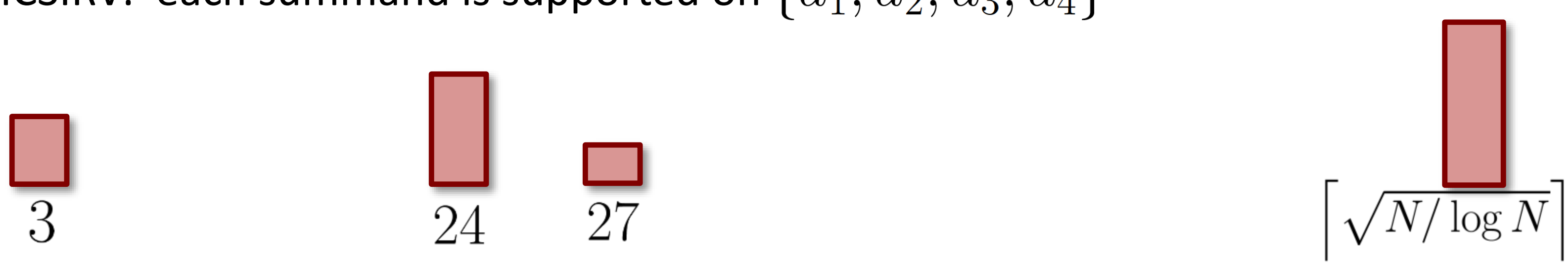
$$\mathbf{S} = \mathbf{X}_1 + \cdots + \mathbf{X}_N$$

where the $\mathbf{X}_i$‘s are independent and each supported on $\{a_1, \ldots, a_k\}$.

- 4-SIIRV: each summand is supported on $\{0, 1, 2, 3\}$



- 4-SICSIRV: each summand is supported on $\{a_1, a_2, a_3, a_4\}$



An easy, but weak, observation about learning SICSIRVs

- Can view a SICSIRV over $\{a_1, \ldots, a_k\}$ as a (degenerate) a_k-SIIRV.

- So by [DDOST13], can learn in time $\mathrm{poly}(a_k, 1/\varepsilon)$.

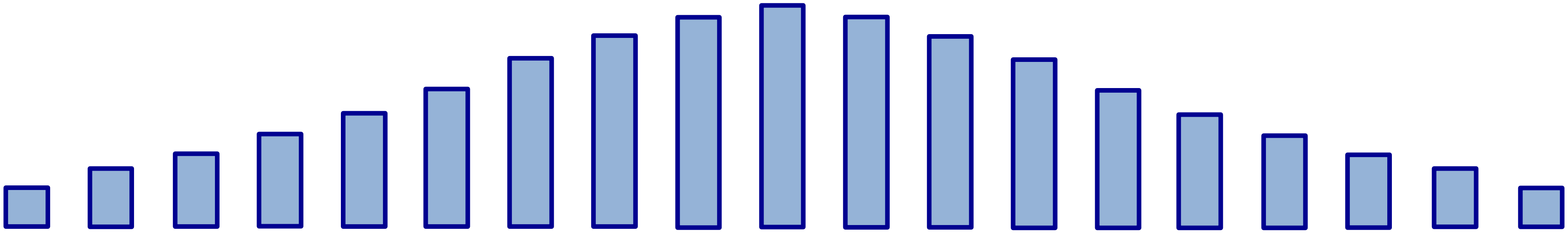
- This may be terrible – we may have $a_k = 2^{2^{2^{\cdot^{\cdot^{\cdot}}}}}$.

Can we do better?

Learning SICSIRVs?

- Support size two is easy: for any set $\{a_1, a_2\}$, a SICSIRV over $\{a_1, a_2\}$ is a scaled and translated Poisson Binomial Distribution (all RV's supported on {0,1}).

$a_1 = 7, a_2 = 9:$ equivalent to 2*(shifted scaled PBD)



Can learn in time poly$(1/\epsilon)$

- What about supports $\{a_1, a_2, a_3\}$ of size three?

First main result: Algorithm for k=3

Theorem: [DLS16] There is an algorithm which, given any support set $\{a_1, a_2, a_3\}$, can learn any unknown SICSIRV over $\{a_1, a_2, a_3\}$ in time poly$(1/\epsilon)$.

Runtime (and sample complexity) independent of N **and of** $\{a_1, a_2, a_3\}$.

Second main result: Algorithm for general k

Theorem: [DLS16] There is an algorithm which, given any support set , ca $\{a_1, a_2, a_3\}$ nknown SICSIRV over in t $\{a_1, a_2, a_3\}$.
$\mathrm{poly}(1/\epsilon)$

Theorem: [DLS16] For any constant $k \geq 4$, there is an algorithm which, given any support set $\{a_1, \ldots, a_k\}$, can learn any unknown SICSIRV over $\{a_1, \ldots, a_k\}$ using

$$\mathrm{poly}(1/\varepsilon) \cdot \log \log a_k$$

samples and $\mathrm{poly}(1/\varepsilon, \log a_k)$ running time.

Improvable? No.

Third main result: Lower bound for k>3

Theorem: [DLS16] There are infinitely many sets $\{a_1, a_2, a_3, a_4\}$ such that any algorithm for learning SICSIRVs over $\{a_1, a_2, a_3, a_4\}$ must use

$$\Omega(\log \log a_4)$$

many samples (for N sufficiently large).

Sharp transition between sets of sizes 3 and 4!

Some ingredients of the algorithm for k=3

Theorem: [DLS16] There is an algorithm which, given any support set , ca$\{a_1, a_2, a_3\}$nknown SICSIRV over in t$\{a_1, a_2, a_3\}$.
poly$(1/\epsilon)$

Theorem: [DLS16] There is an algorithm which, given any support set , ca $\{a_1, a_2, a_3\}$ nknown SICSIRV over in t $\{a_1, a_2, a_3\}$.
poly$(1/\epsilon)$

Without loss of generality, assume that the support set is $\{0, p, q\}$ where $\gcd(p, q) = 1.$

With (significant) loss of generality, assume each summand is either supported on $\{0, p\}$ or on $\{0, q\}$.

In other words, the target distribution is $p\mathbf{X}^{(p)} + q\mathbf{X}^{(q)}$ where $\mathbf{X}^{(p)}, \mathbf{X}^{(q)}$ are independent Poisson Binomial distributions.

What does $p\mathbf{X}^{(p)} + q\mathbf{X}^{(q)}$ look like?

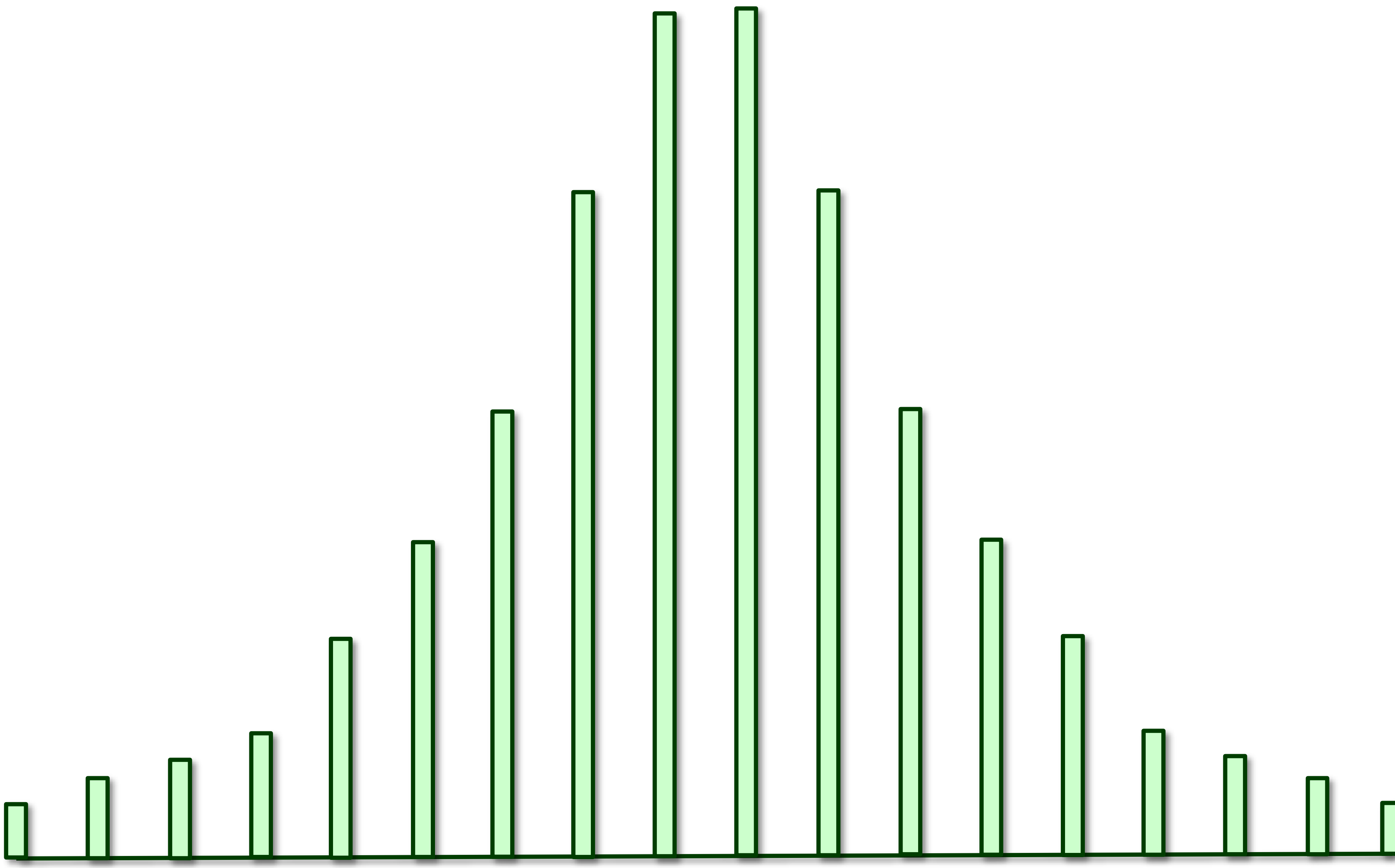
Assume that $\mathrm{Var}[\mathbf{X}^{(p)}], \mathrm{Var}[\mathbf{X}^{(q)}] \geq \mathrm{poly}(1/\varepsilon)$

Assume that $\mathrm{Var}[p\mathbf{X}^{(p)}] \geq \mathrm{Var}[q\mathbf{X}^{(q)}]$

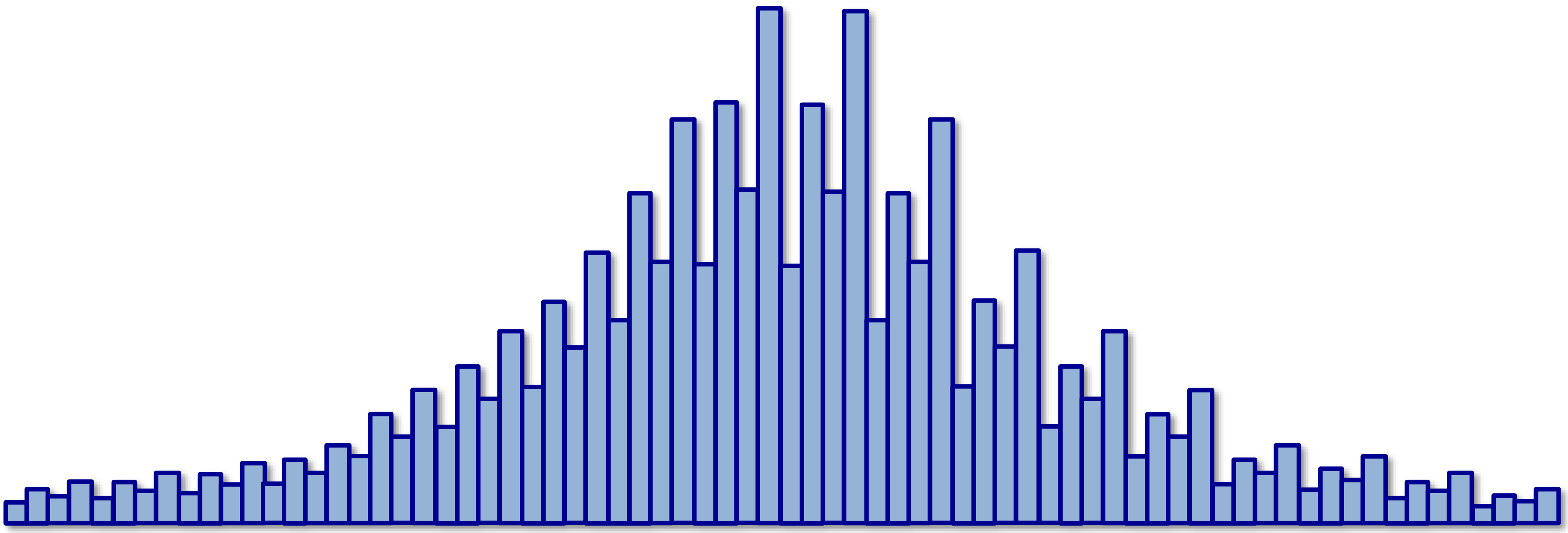
Informal Lemma: The random variable

$$\mathbf{Z} = p\mathbf{X}^{(p)} + q\mathbf{X}^{(q)}$$

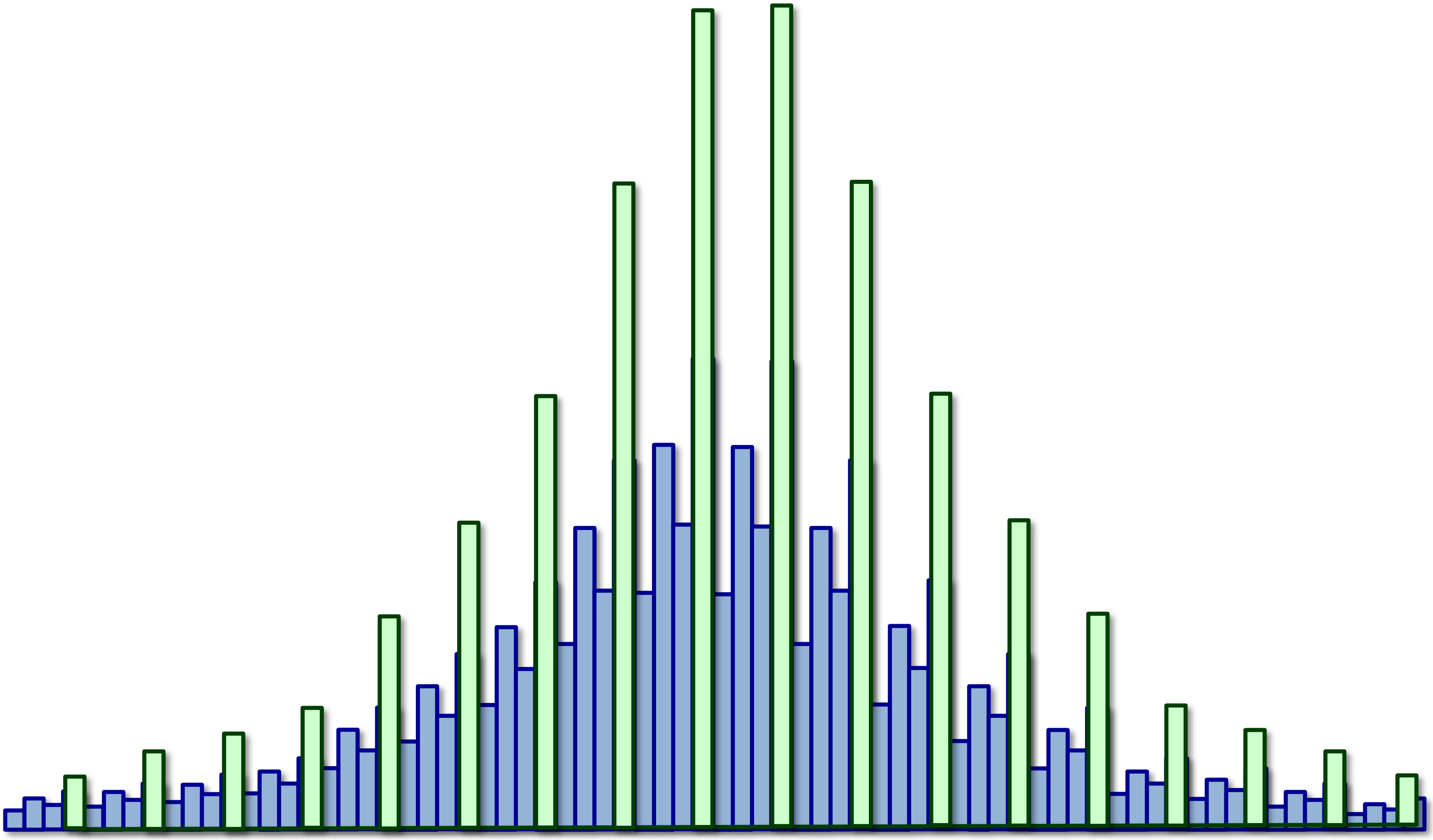
looks like a discretized Gaussian if you *blur your eyes at the scale of p.*



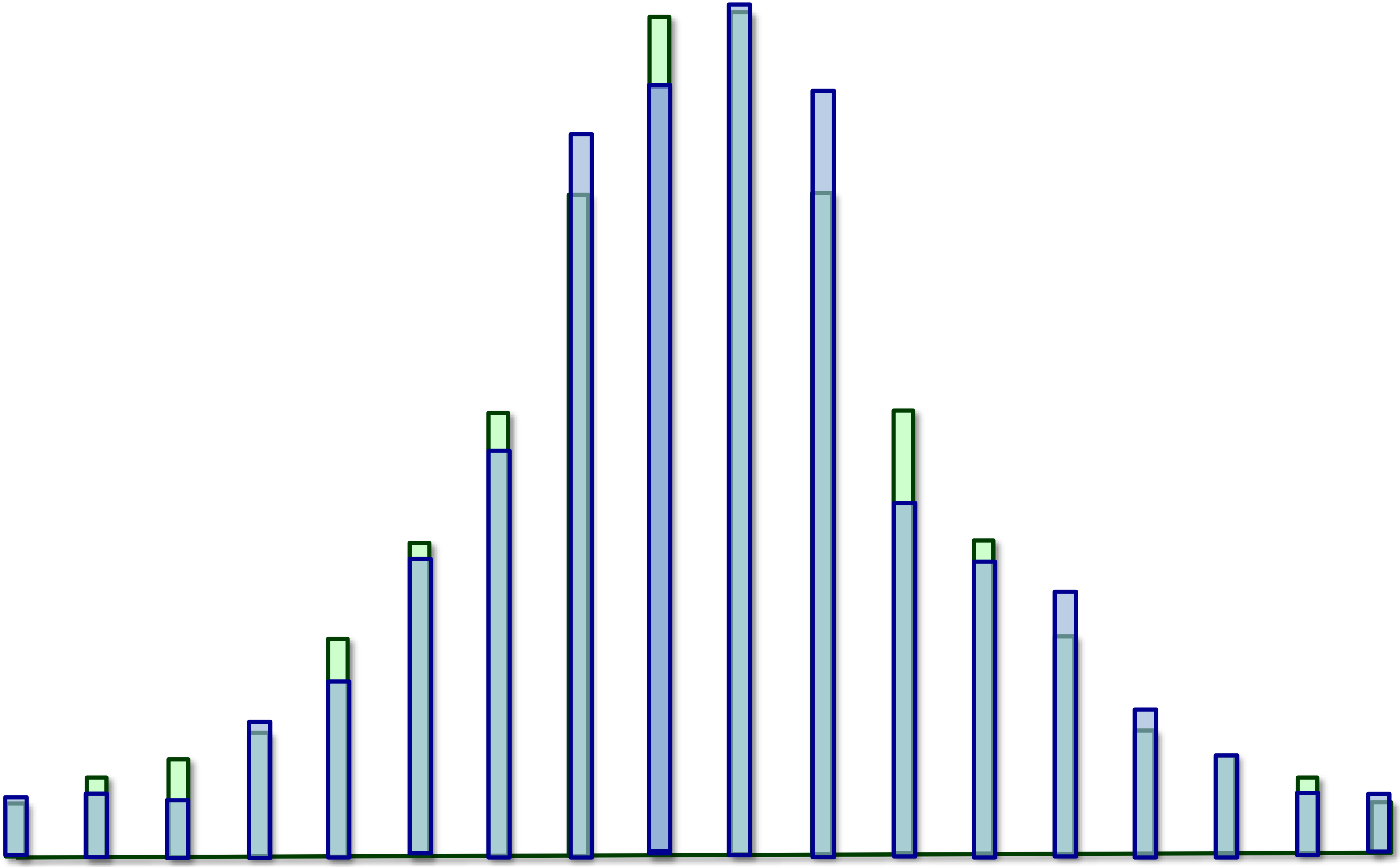
p-scaled discretized Gaussian



The distribution $p\mathbf{X}^{(p)} + q\mathbf{X}^{(q)}$



Total variation distance between the distributions may be large.



But if you round each distribution to the nearest multiple of p, they are close to each other in total variation distance.

What does $p\mathbf{X}^{(p)} + q\mathbf{X}^{(q)}$ look like?

Informal Lemma: The random variable $\mathbf{Z} = p\mathbf{X}^{(p)} + q\mathbf{X}^{(q)}$ looks like a discretized Gaussian if you *blur your eyes at the scale of p.*

Need to understand: what does $\mathbf{Z}$ look like mod p?

→ To answer this, we need to study the structure of $q\mathbf{X}^{(q)}$ mod p.

“Two” cases

Let σ_q denote $\sqrt{\mathrm{Var}[\mathbf{X}^{(q)}]}$

Two cases:

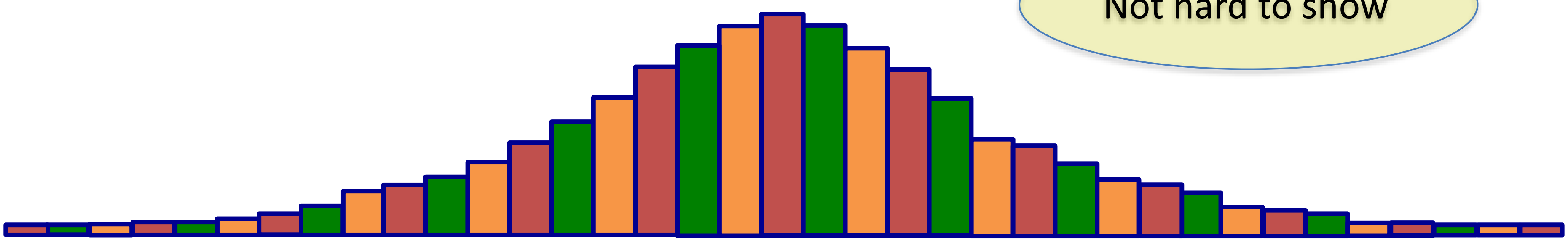
- $\mathrm{Var}[\mathbf{X}^{(q)}]$ big: $\sigma_q \gg p/\varepsilon$
- $\mathrm{Var}[\mathbf{X}^{(q)}]$ small: $\sigma_q \ll \varepsilon \cdot p$

(We’ll not come back to the missing case later…)

First case: $\sigma_q \gg p/\varepsilon$

Lemma: If $\sigma_q \gg p/\varepsilon$, then $q\mathbf{X}^{(q)}$ is close to uniformly distributed in $\mathbb{Z}_p$.

Not hard to show



$:= 0 \mod 3$

$:= 1 \mod 3$

$:= 2 \mod 3$

All residue classes modulo 3 are roughly equidistributed.

First case: $\sigma_q \gg p/\varepsilon$

Lemma: If $\sigma_q \gg p/\varepsilon$, then $q\mathbf{X}^{(q)}$ is close to uniform over $\mathbb{Z}_p$.

Lemma: If $q\mathbf{X}^{(q)}$ is uniform over $\mathbb{Z}_p$, then

$$\mathbf{Z} = p\mathbf{X}^{(p)} + q\mathbf{X}^{(q)}$$

is close to a discretized Gaussian (with no scaling).

Intuition: $q\mathbf{X}^{(q)}$ "fills in the gaps" between multiples of p

Proof uses a generalization of the notion of *shift-invariance* from probability theory (a measure of smoothness of probability distributions).

Second case: $\sigma_q \ll \varepsilon \cdot p$

Informal Lemma: If $\sigma_q \ll \varepsilon \cdot p$, then can learn $q\mathbf{X}^{(q)}$ given draws from $\mathbf{Z} = p\mathbf{X}^{(p)} + q\mathbf{X}^{(q)}$.

Example: Suppose p=1,000,000,000, q=1, $\sigma_q = 1000$.

Draws from $\mathbf{Z} = p\mathbf{X}^{(p)} + q\mathbf{X}^{(q)}$

8,729,341,007,000,497
8,512,223,006,998,725
8,485,874,007,001,301
8,629,534,007,000,837
8,553,897,006,999,134

Key insight: take samples mod p!

Draws from $q\mathbf{X}^{(q)} = \mathbf{X}^{(q)}$

7,000,497
6,998,725
7,001,301
7,000,837
6,999,134

(easy to learn the PBD $\mathbf{X}^{(q)}$ from these draws)

In general when q > 1: multiply these values by q^{-1} mod p to get draws from $\mathbf{X}^{(q)}$.

A peek at the general-k algorithm

General k: consider target $a_1\mathbf{X}^{(1)} + \cdots + a_k\mathbf{X}^{(k)}$.

Let us assume $a_k\mathbf{X}^{(k)}$ contributes plurality of the variance. For each $1 \leq i < k$, two possibilities:

1. $\sqrt{\mathsf{Var}[\mathbf{X}^{(i)}]} \geq a_k/\epsilon$: The component "gets absorbed" in $a_k\mathbf{X}^{(k)}$.

2. $\sqrt{\mathsf{Var}[\mathbf{X}^{(i)}]} \leq a_k/\epsilon$: Up to a multiplicative factor of 1 + ε, there are $\log(a_k)$ possibilities.

Hypothesis testing : $\log\log(a_k)$ samples.

End of the algorithms part

A word about the lower bound

Theorem: There are infinitely many sets $\{0, p, q, r\}$ such that any algorithm for learning SICSIRVs over $\{0, p, q, r\}$ must use

$$\Omega(\log \log r)$$

many samples (for N sufficiently large).

(a) Choose $p = 1$.

(b) Choice of q and r exploits delicate properties of continued fractions.

Rational approximations of continued fractions

$$\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cdots}}} = \frac{1}{\phi} \leftarrow \text{Golden Ratio}$$

Let q_ℓ / r_ℓ be the ℓ^{th} convergent of this continued fraction. Then,

$$\left| \frac{q_\ell}{r_\ell} - \frac{1}{\phi} \right| = \Theta\left(\frac{1}{r_\ell^2}\right)$$

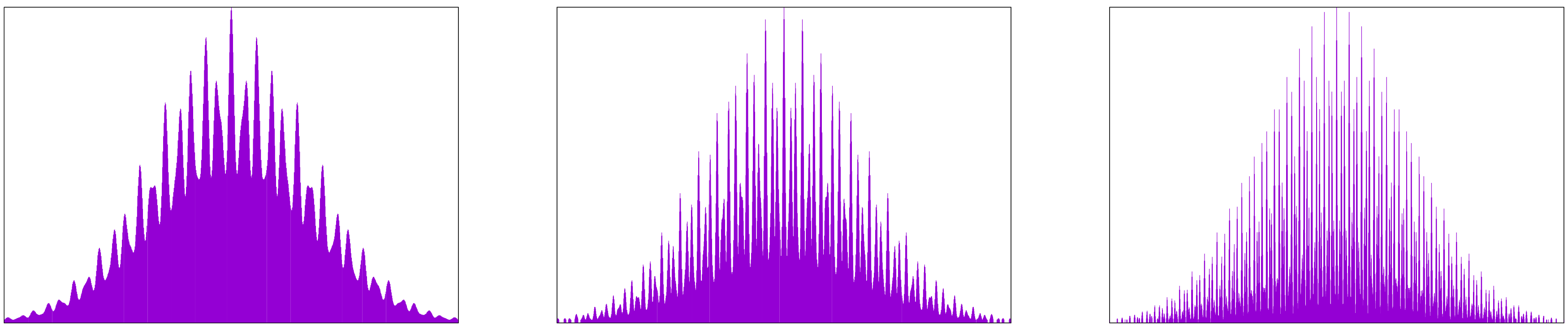
Rational approximations of continued fractions

$$\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cdots}}} = \frac{1}{\phi} \leftarrow \text{Golden Ratio}$$

Let q_ℓ / r_ℓ be the ℓ^{th} convergent of this continued fraction. Then,

$$\left| \frac{q_\ell}{r_\ell} - \frac{1}{\phi} \right| = \Theta\left(\frac{1}{r_\ell^2} \right)$$

Picture aided proof

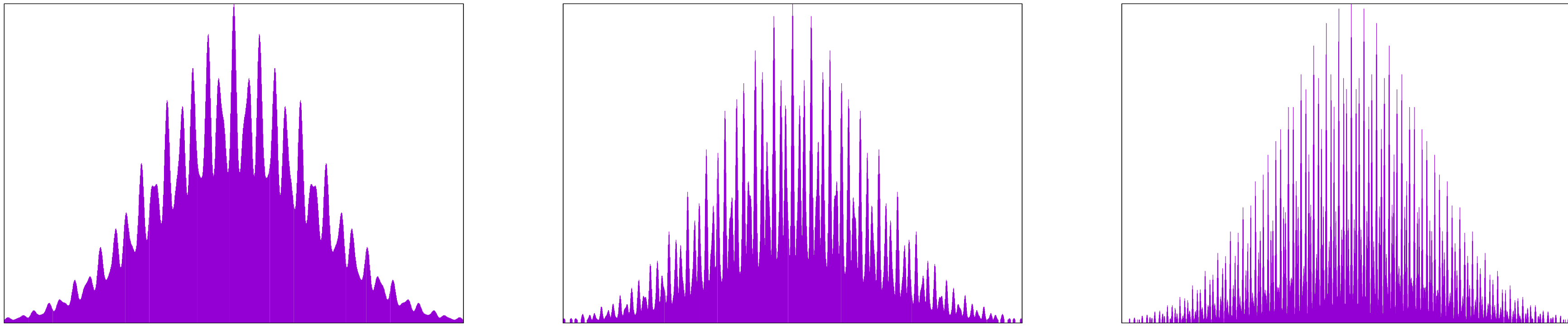


(a) (b) (c)

We construct a family of $\Omega(\log r)$ SICSIRVs over the set $\{0,1,q,r\}$ such that

(1) All these distributions look like Gaussians at *the scale of r.*
(2) The "mod r" structure is different among these distributions.
(3) The peak-valley structure becomes finer as we go from (a) to (c).
(4) In each distribution, nearby peaks and valleys have mass ratio at most (constant)

Picture aided proof



(a) (b) (c)

We thus obtain $\Omega(\log r)$ SICSIRVs over the set $\{0, 1, q, r\}$ such that

(1) ℓ_1 distance between any two of these distributions is > (some constant).

(2) KL-divergence between any two of these distributions is at most (some constant).

This is sufficient for us to apply Fano's inequality and obtain a $\Omega(\log \log r)$ lower bound.

Last results:

Learning SICSIRVs when the support set is **unknown**

We have assumed so far that the learning algorithm is given $\{a_1, \ldots, a_k\}$.

Let's relax this assumption, and assume the algorithm is only given an *upper bound* $a_{\max} \geq a_1, \ldots, a_k$.

What happens then?

For general k: Can try all possible $(a_1, \ldots, a_k) \in [a_{\max}]^k$. Test all $(a_{\max})^k$ resulting hypotheses, choose best one.

- Fact: Can find an $O(\varepsilon)$-good hypothesis from pool of M hypotheses containing an ε-good one, using $\log(M) \cdot \mathrm{poly}(1/\varepsilon)$ samples and $\mathrm{poly}(M, 1/\varepsilon)$ runtime.

So for $k \geq 3$, can learn using $\log(a_{\max}) \cdot \mathrm{poly}(1/\varepsilon)$ samples.

This was loglog when support was known…

For $k \geq 3$, can learn using $\log(a_{\max}) \cdot \mathrm{poly}(1/\varepsilon)$ samples

This is the best that can be done for unknown support, even for k=3:

Theorem: [DLS16] There are infinitely many values $a_{\max}$ such that any algorithm for learning SICSIRVS over unknown $a_1, a_2, a_3 \leq a_{\max}$ must use

$$\Omega(\log a_{\max})$$

many samples (for N sufficiently large).

For $k \geq 3$, can learn using $\log(a_{\max}) \cdot \mathrm{poly}(1/\varepsilon)$ samples

This is the best that can be done for unknown support, even for k=3:

Theorem: [DLS16] There are infinitely many values $a_{\max}$ such that any algorithm for learning SICSIRVS over unknown $a_1, a_2, a_3 \leq a_{\max}$ must use

$$\Omega(\log a_{\max})$$

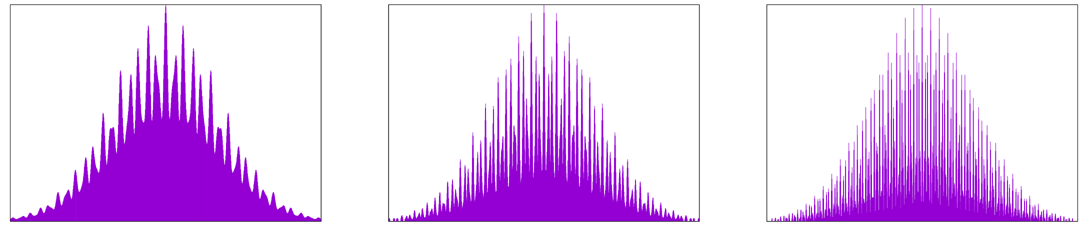
many samples (for N sufficiently large).

Uses equidistribution results from number theory.

Summary

- SICSIRV = **S**um of **I**ndependent **C**ommonly **S**upported **I**nteger **R**andom **V**ariables
- Good understanding of sample, runtime complexity of learning SICSIRVS over $\{a_1, \ldots, a_k\}$ for all k, both in known-support and unknown-support settings
- Independence is really powerful
- Future work: beyond independence?

Thank you!



(a) (b) (c)