A New Approach to Distribution Testing

Daniel M. Kane

Department of Mathematics / Department of Computer Science and Engineering
University of California, San Diego
dakane@ucsd.edu

September 8th, 2016

Joint work with
Ilias Diakonikolas (USC)

1. Introduction
2. Uniformity Testing
3. L^2 Testers
4. Testing Closeness to Known Distribution
5. Testing Closeness to Unknown Distribution
6. Testing Independence
7. Instance Optimality
8. Other Applications and Future Work

Distribution Testing

Basic statistics question: Given a bunch of independent samples from a probability distribution (or perhaps from several), determine whether or not it has some property.

Distribution Testing

Basic statistics question: Given a bunch of independent samples from a probability distribution (or perhaps from several), determine whether or not it has some property.

Example properties:

- p is uniform.
- $p = q$.
- The coordinates of p are independent.

History

- Hypothesis testing introduced by Pearson in 1899.
- Classical problem in statistics [Neyman-Pearson33, Lehman-Romano05]
- Recently taken up by the TCS community [Goldreich-Ron00, BFFKRW FOCS00/JACM13]

Closeness

Problem

Cannot distinguish between p with property and arbitrarily close p' without.

Closeness

Problem

Cannot distinguish between p with property and arbitrarily close p' without.

Solution

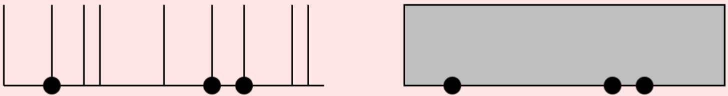
Distinguish between

- p has property.
- p is far (usually in L^1) from any distribution with property.

Continuous

Problem

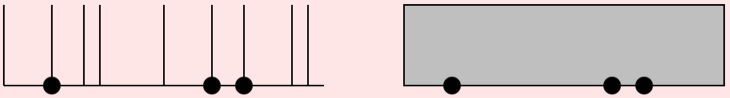
Cannot distinguish between continuous distribution and discrete distribution with large random support.



Continuous

Problem

Cannot distinguish between continuous distribution and discrete distribution with large random support.



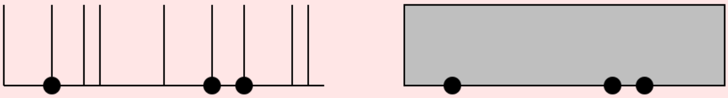
Solutions

- Consider only *structured*, low-complexity distributions.
- Consider only discrete distributions on finite domain.

Continuous

Problem

Cannot distinguish between continuous distribution and discrete distribution with large random support.



Solutions

- Consider only *structured*, low-complexity distributions.
- Consider only discrete distributions on finite domain.

We will focus on the latter.

Notation

- Distributions p, q on $[n] := \{1, 2, \ldots, n\}$.
- $p_i := \Pr(p = i), q_i := \Pr(q = i)$.
- Question like: distinguish between
  - $p = q$
  - $\|p - q\|_1 \geq \epsilon$

  with at least $2/3$ probability of success.

Goal

Want:

- Number of samples information-theoretically optimal.
- Runtime polynomial (or even linear) in number of samples.

Simple Question

Distinguish between:

- p is the uniform distribution.
- $\|p - U_n\|_1 = \Omega(1)$.

Stats 101 Answer

- Take m samples from p.
- Let X_i be from bin i.
- Note $X_i \approx \text{Gaussian}$.
- Compute

$$Z := \sum_{i=1}^{n} (X_i - m/n)^2$$

and compare to appropriate χ^2 distribution.

Stats 101 Answer

- Take m samples from p.
- Let X_i be from bin i.
- Note $X_i \approx \mathrm{Gaussian}$.
- Compute
$$Z := \sum_{i=1}^{n} (X_i - m/n)^2$$
and compare to appropriate χ^2 distribution.

Problem: Need $\Omega(n)$ samples for Gaussian approximation.

Improvement

Observation [Goldreich-Ron]

Taking samples from the uniform distribution gives fewer expected collisions than from any other distribution.

Improvement

Observation [Goldreich-Ron]

Taking samples from the uniform distribution gives fewer expected collisions than from any other distribution.

Algorithm

- Take m samples.
- Count collisions.
- Compare to number expected under uniform distribution.

Improvement

Observation [Goldreich-Ron]

Taking samples from the uniform distribution gives fewer expected collisions than from any other distribution.

Algorithm

- Take m samples.
- Count collisions.
- Compare to number expected under uniform distribution.

Takes about $\sqrt{n}$ samples to get collision. Sample complexity $O(\sqrt{n})$.

Quadratic Testers

- Both testers use quadratic test statistics.
- Very natural thing to do.
- As we will see quite powerful.

Problem

- Distributions p, q on $[n]$.
- Take m samples from each.
- Distinguish between
  - $p = q$
  - p far from q

Simple Tester

- X_i number of samples from p in i^{th} bin.
- Y_i number of samples from q in i^{th} bin.
- Test statistic

$$Z = \sum_{i=1}^{n} (X_i - Y_i)^2.$$

Poissonization

Trick

Take $\mathrm{Poi}(m)$ samples from p and $\mathrm{Poi}(m)$ samples from q.

Poissonization

Trick

Take $\mathrm{Poi}(m)$ samples from p and $\mathrm{Poi}(m)$ samples from q.

- Makes X_i, Y_i independent.
- $X_i \sim \mathrm{Poi}(mp_i)$, $Y_i \sim \mathrm{Poi}(mq_i)$
- Likely doesn't change total number of samples by much.

Expectation

Have

$$\mathbb{E}[(X_i - Y_i)^2] = m^2(p_i - q_i)^2 + m(p_i + q_i).$$

Expectation

Have

$$\mathbb{E}[(X_i - Y_i)^2] = m^2(p_i - q_i)^2 + m(p_i + q_i).$$

Fix:

$$Z = \sum_{i=1}^{n}(X_i - Y_i)^2 - X_i - Y_i.$$

Expectation

Have

$$\mathbb{E}[(X_i - Y_i)^2] = m^2(p_i - q_i)^2 + m(p_i + q_i).$$

Fix:

$$Z = \sum_{i=1}^{n} (X_i - Y_i)^2 - X_i - Y_i.$$

Then

$$\begin{aligned} \mathbb{E}[Z] &= m^2\|p-q\|_2^2 \\ \mathrm{Var}(Z) &= O(m^3\|p-q\|_2^2\|p+q\|_2 + m^2\|p+q\|_2^2) \end{aligned}$$

L^2 Tester

L^2 Tester [Chan-Diakonikolas-Valiant-Valiant]

There is a tester that distinguishes between $p = q$ and $\|p - q\|_2^2 \geq \epsilon^2$ in expected $O(\|p + q\|_2/\epsilon^2)$ samples.

L^2 Tester

L^2 Tester [Chan-Diakonikolas-Valiant-Valiant]

There is a tester that distinguishes between $p = q$ and $\|p - q\|_2^2 \geq \epsilon^2$ in expected $O(\|p + q\|_2/\epsilon^2)$ samples.

Note

By first testing if $\|p\|_2 \approx \|q\|_2$, can reduce to $O(\min(\|p\|_2, \|q\|_2)/\epsilon^2 + \min(1/\|p\|_2, 1/\|q\|_2))$ samples.

L^2 Tester

L^2 Tester [Chan-Diakonikolas-Valiant-Valiant]

There is a tester that distinguishes between $p = q$ and $\|p - q\|_2^2 \geq \epsilon^2$ in expected $O(\|p + q\|_2/\epsilon^2)$ samples.

Note

By first testing if $\|p\|_2 \approx \|q\|_2$, can reduce to $O(\min(\|p\|_2, \|q\|_2)/\epsilon^2 + \min(1/\|p\|_2, 1/\|q\|_2))$ samples.

Note II

In fact, this tester is *tolerant*. It can distinguish between $\|p - q\|_2^2 \leq \epsilon^2/2$ and $\|p - q\|_2^2 \geq \epsilon^2$.

Main New Idea

Solve *all* problems by reducing to this as a black box.

Problem

Compare p to explicitly known distribution.

- Given explicit distribution q on $[n]$.
- Given m samples from p on $[n]$.
- Distinguish between
  - $p = q$
  - $\|p - q\|_1 \geq \epsilon$.

Using L^2 Tester

Need to distinguish between $p = q$ and $\|p - q\|_2^2 \geq \epsilon^2/n$.

- Simulate samples from q.
- Takes $O(n\|q\|_2/\epsilon^2)$ samples.

Using L^2 Tester

Need to distinguish between $p = q$ and $\|p - q\|_2^2 \geq \epsilon^2/n$.

- Simulate samples from q.
- Takes $O(n\|q\|_2/\epsilon^2)$ samples.
- If q near uniform, this is $O(\sqrt{n}/\epsilon^2)$, which is optimal.
- If $\|q\|_2$ is large, test statistic has too much variance.

Using L^2 Tester

Need to distinguish between $p = q$ and $\|p - q\|_2^2 \geq \epsilon^2/n$.

- Simulate samples from q.
- Takes $O(n\|q\|_2/\epsilon^2)$ samples.
- If q near uniform, this is $O(\sqrt{n}/\epsilon^2)$, which is optimal.
- If $\|q\|_2$ is large, test statistic has too much variance.

Question

How do we deal with this?

Previous Work

- **[Batu-Fortnow-Kumar-Rubinfeld-Smith-White '00]:** Split bins into buckets in which q is near-uniform.
- **[Valiant-Valiant '14]:** Modify the test statistic to give less weight to heavy bins.

Our Technique

Divide i^{th} bin into $\lceil nq_i \rceil$ equally sized bins. Have new distributions p', q'.

Facts

- $\|p' - q'\|_1 = \|p - q\|_1$.
- Can sample from p'.
- New domain size $O(n)$.
- q' approximately uniform.

Our Technique

Divide i^{th} bin into $\lceil nq_i \rceil$ equally sized bins. Have new distributions p', q'.

Facts

- $\|p' - q'\|_1 = \|p - q\|_1$.
- Can sample from p'.
- New domain size $O(n)$.
- q' approximately uniform.

Requires $O(\sqrt{n}/\epsilon^2)$ samples.

Reduction

Essentially, we reduced to the case where $q_i = O(1/n)$ for all i.

Reduction

Essentially, we reduced to the case where $q_i = O(1/n)$ for all i.

Recent improvement by Goldreich shows how to reduce to $q = \mathrm{Uniform}$.

Unknown q

- What happens if instead q is *unknown* and we are only given sample access?

Unknown q

- What happens if instead q is *unknown* and we are only given sample access?
- We no longer know how to break bins up or reweight Z.

Unknown q

- What happens if instead q is *unknown* and we are only given sample access?
- We no longer know how to break bins up or reweight Z.
- Testing is actually *harder*. There is a lower bound of

$$\Omega(\max(\sqrt{n}/\epsilon^2, n^{2/3}/\epsilon^{4/3}))$$

samples by Chan-Diakonikolas-Valiant-Valiant.

Previous Work

- **[Batu-Fisher-Fortnow-Kumar-Rubinfeld-White '00]:** Learn the heavy bins of q, and run L^2 tester on light bins. (Gives $O(n^{2/3}\log(n)/\epsilon^{8/3})$ samples)
- **[Valiant '08]:** Learn heavy bins of q and see if higher moments of p and q on low bins match. (Gives $O(n^{2/3})$ samples for constant ϵ)
- **[Chan-Diakonikolas-Valiant-Valiant '14]:** Uses different test statistic

$$\sum_i \frac{(X_i - Y_i)^2 - X_i - Y_i}{X_i + Y_i}.$$

Sample optimal.

Idea

- Need to divide heavier bins into more pieces.
- How to detect heavy bins?

Idea

- Need to divide heavier bins into more pieces.
- How to detect heavy bins?
- Use samples.

Our Technique

Take $\mathrm{Poi}(k)$ samples from q. If a_i samples from bin i, divide into $a_i + 1$ pieces.

Our Technique

Take $\mathrm{Poi}(k)$ samples from q. If a_i samples from bin i, divide into $a_i + 1$ pieces.

$$\|q'\|_2^2 = \sum_i (a_i + 1)\left(\frac{q_i}{a_i+1}\right)^2 = \sum_i \frac{q_i^2}{a_i+1}.$$

Our Technique

Take $\mathrm{Poi}(k)$ samples from q. If a_i samples from bin i, divide into $a_i + 1$ pieces.

$$\|q'\|_2^2 = \sum_i (a_i + 1)\left(\frac{q_i}{a_i + 1}\right)^2 = \sum_i \frac{q_i^2}{a_i + 1}.$$

$$\mathbb{E}[\|q'\|_2^2] = \sum_i q_i^2 \mathbb{E}[1/(a_i+1)] = \sum_i O(q_i^2/(kq_i)) = O(1/k).$$

Algorithm

Algorithm

- Let $k = \min(n, n^{2/3}/\epsilon^{4/3})$.
- Take $\mathrm{Poi}(k)$ samples from q, and divide bins based on samples.
- Run L^2 tester to see if $p' = q'$ or $\|p' - q'\|_1 \geq \epsilon$.

Algorithm

Algorithm

- Let $k = \min(n, n^{2/3}/\epsilon^{4/3})$.
- Take $\mathrm{Poi}(k)$ samples from q, and divide bins based on samples.
- Run L^2 tester to see if $p' = q'$ or $\|p' - q'\|_1 \geq \epsilon$.

Likely have

- $O(n)$ bins.
- $\|q'\|_2 = O(1/\sqrt{k})$.

Samples Needed

$$O(k + nk^{-1/2}\epsilon^{-2}) = O(\max(\sqrt{n}/\epsilon^2, n^{2/3}/\epsilon^{4/3})).$$

Samples Needed

$$O(k + nk^{-1/2}\epsilon^{-2}) = O(\max(\sqrt{n}/\epsilon^2, n^{2/3}/\epsilon^{4/3})).$$

This also works if you can take *unequal* numbers of samples from the two distributions.

- $O(m)$ samples from p
- $O(k + m)$ samples from q
- Where $m = O(\sqrt{n}/\epsilon^2 + nk^{-1/2}/\epsilon^2)$.

Problem

p a distribution on $[n] \times [m]$ for $n \geq m$. Given samples from p distinguish between cases:

- The coordinates of p are independent.
- p is at least ϵ-far from any distribution with independent coordinates.

Previous Work

Upper bounds

- **[Batu-Fisher-Fortnow-Kumar-Rubinfeld-White '01]:** $\tilde{O}(n^{2/3}m^{1/3}\mathrm{poly}(1/\epsilon))$.
- **[Acharya-Daskalakis-Kamath '15]:** $O(n/\epsilon^2)$ for $n = m$.

Previous Work

Upper bounds

- **[Batu-Fisher-Fortnow-Kumar-Rubinfeld-White '01]:** $\tilde{O}(n^{2/3}m^{1/3}\mathrm{poly}(1/\epsilon))$.
- **[Acharya-Daskalakis-Kamath '15]:** $O(n/\epsilon^2)$ for $n = m$.

Lower bounds

- **[Levi-Ron-Rubinfeld '11]:** $\tilde{\Omega}(n^{2/3}m^{1/3})$ for constant error.
- **[Diakonikolas-K '16]:** $\Omega(\max(n^{2/3}m^{1/3}/\epsilon^{4/3}, \sqrt{nm}/\epsilon^2))$

Our Technique

- Compare p to $q = p_1 \times p_2$.
- Need to flatten q. Do by flattening p_1, p_2.

Algorithm

Algorithm

- Take $\mathrm{Poi}(m)$ samples from p_2, use to subdivide bins of $[m]$.
- Let $k = \min(n, n^{2/3}m^{1/3}/\epsilon^{4/3})$.
- Take $\mathrm{Poi}(k)$ samples from p_1, use to subdivide bins of $[n]$.
- Use L^2 tester to distinguish $p' = q'$ or $\|p' - q'\|_1 \geq \epsilon$.

Analysis

Probably have:

- New array $O(n) \times O(m)$.
- $\|p_1'\|_2 = O(1/\sqrt{k})$, $\|p_2'\|_2 = O(1/\sqrt{m})$.

Analysis

Probably have:

- New array $O(n) \times O(m)$.
- $\|p_1'\|_2 = O(1/\sqrt{k})$, $\|p_2'\|_2 = O(1/\sqrt{m})$.

Samples needed:

$$O(k + m + nmk^{-1/2}m^{-1/2}/\epsilon^2)$$
$$=O(\max(n^{2/3}m^{1/3}/\epsilon^{4/3}, \sqrt{nm}/\epsilon^2)).$$

Analysis

Probably have:

- New array $O(n) \times O(m)$.
- $\|p'_1\|_2 = O(1/\sqrt{k})$, $\|p'_2\|_2 = O(1/\sqrt{m})$.

Samples needed:

$$O(k + m + nmk^{-1/2}m^{-1/2}/\epsilon^2)$$
$$= O(\max(n^{2/3}m^{1/3}/\epsilon^{4/3}, \sqrt{nm}/\epsilon^2)).$$

Optimal!

Worst Case

Testing identity to a known distribution requires $O(\sqrt{n}/\epsilon^2)$ samples, but only for worst-case q. This lower bound is not hard to prove for q uniform or nearly uniform, but for other q you can often do better.

Instance Optimality

[Valiant-Valiant '14] provide an *instance optimal* tester. That is a tester that for each q gives a tester with the fewest number of samples *for that q*. The complexity is (usually) $\Theta(\|q\|_{2/3}/\epsilon^2)$.

Instance Optimality

[Valiant-Valiant '14] provide an *instance optimal* tester. That is a tester that for each q gives a tester with the fewest number of samples *for that* q. The complexity is (usually) $\Theta(\|q\|_{2/3}/\epsilon^2)$.

The basic technique involves a careful reweighting of the L^2 tester.

Our Results

Using the L^2 tester as a black box, we can get within polylogarithmic factors.

Algorithm

- Divide bins into (logarithmically many) categories based on $\lfloor \log(q_i) \rfloor$.
- Test that p assigns approximately the right mass to each category.
- For each category, C, test whether $(p|C) = (q|C)$ or $\|(p|C) - (q|C)\|_1 \geq \epsilon / \mathrm{Pr}(C)\mathrm{polylog}(n/\epsilon)$.

Analysis

Testing over categories is easy. Consider a single category C.

- All bins mass $\Theta(x)$.
- m total bins.
- $\Pr(C) = \Theta(mx)$.

Analysis

Testing over categories is easy. Consider a single category C.

- All bins mass $\Theta(x)$.
- m total bins.
- $\Pr(C) = \Theta(mx)$.
- Need $\mathrm{polylog}(n/\epsilon)\sqrt{m}/(\epsilon/(mx))^2 = \mathrm{polylog}(n/\epsilon)m^{5/2}x^2/\epsilon^2$ samples from $p|C$.
- Need $\mathrm{polylog}(n/\epsilon)m^{3/2}x/\epsilon^2$ samples from p.

Analysis

$$\|q\|_{2/3} \approx \left(\max_C (mx^{2/3})\right)^{3/2} = \max_C (m^{3/2}x).$$

Sample complexity $\mathrm{polylog}(n/\epsilon)\|q\|_{2/3}/\epsilon^2$.

Analysis

$$\|q\|_{2/3} \approx \left(\max_C (m x^{2/3})\right)^{3/2} = \max_C (m^{3/2} x).$$

Sample complexity $\mathrm{polylog}(n/\epsilon)\|q\|_{2/3}/\epsilon^2$.
Correct up to polylogarithmic factors.

Unknown q

Perhaps more surprisingly, we can do almost as well without knowing q ahead of time.

Idea

- Take m samples from q.
- Divide bins into categories based of $\lfloor \log(\mathrm{samples}) \rfloor$.
- Check that p assigns roughly same mass to categories.
- Test whether restriction of p to categories approximates q.

Analysis

- Bins with more than $1/m$ mass sorted into category with other bins of approximately the same size.
- On these categories looks like instance optimal tester.
- Remaining bin uses L^2 tester.

Analysis

- Bins with more than $1/m$ mass sorted into category with other bins of approximately the same size.
- On these categories looks like instance optimal tester.
- Remaining bin uses L^2 tester.

Complexity

$$\mathrm{polylog}(n/\epsilon) \min_m \left(m + \|q\|_{2/3}/\epsilon^2 + \|q^{<1/m}\|_2 \|q^{<1/m}\|_0/\epsilon^2 \right).$$

Discussion

- When ϵ small $\tilde{O}(\|q\|_{2/3}/\epsilon^2)$.

Discussion

- When ϵ small $\tilde{O}(\|q\|_{2/3}/\epsilon^2)$.
- Taking $m = \min(n, n^{2/3}/\epsilon^{4/3})$ get

$$\tilde{O}(\max(\sqrt{n}/\epsilon^2, n^{2/3}/\epsilon^{4/3})).$$

- Only this bad when $\approx m$ bins with mass $\approx 1/m$ and $\approx n$ bins of mass $\approx 1/n$.

Instance Optimal for Unknown q

Unfortunately, there is no way to have instance optimal when q is unknown since different q do not give rise to different problems.

Instance Optimal for Unknown q

Unfortunately, there is no way to have instance optimal when q is unknown since different q do not give rise to different problems.

Can find algorithms that work better with certain q or better with q with certain structure, but you need to choose which structure to take advantage of. What the "right" notion is here is still an open problem.

Other Applications

We also get (nearly) optimal results for:

- Independence testing in higher dimensions.
- Properties of collections of distributions.
- Testing histograms.
- Testing with Hellinger metric.

Future Directions

- Structured distributions.
  Active area (especially for high-dimensional distributions).
- Correct probability of error.
  [Diakonikolas-Gouleakis-Peebles-Price '16] give correct result for identity testing.
- Optimal constants.
  Some work by [Huang-Meyn '14]
- Beyond worst case analysis.