



Testing random variables for independence and identity

Avi Kama

Sublinear time algorithms seminar

Jan 6, 2009

On our talk today

- Background
 - Random variables
 - Independence
 - Closeness
- Introduction of the problem to solve
- More background
 - Restriction
 - Coarsening
- Tools used
 - Bucketing
- Algorithms



Background

The basics

Random variables

- We distinguish between 2 types of given random variables:
 - *Explicit variable* – Given a value i , returns the probability $X(i)$
 - *Black-box variable* – We may sample X independently. Given X , we can retrieve as many independent samples i from X as we want. **X 's distribution is unknown!**

Independence

- Say A is a distribution over $S \times T$
- We say A is independent, if A 's projections over S and T are independent

- $A = (\pi_1 A) \times (\pi_2 A)$, Where $\pi_1 A$ is A projected over S , and $\pi_2 A$ is A projected over T

Notation: $A(i, j)$ is the probability of
(i, j) in A

- All $A(i, j) = (\pi_1 A)(i) \times (\pi_2 A)(j)$

Closeness

- We say that A is ε -independent, if it is ε -close to an independent distribution, B
- That is:
 - $|A - B| \leq \varepsilon$, In the L_1 -Norm
 - L_1 -Norm – Statistical distance. Sum-up probability diffs for all values
- We can show that if $|A - B| \leq \varepsilon/3$, and B is independent, Than:
 - $|A - (\pi_1 A) \times (\pi_2 A)| \leq \varepsilon$



Problem

What we are about to solve

Problem

- **Input:**
 - Black-box distribution A , over $[n] \times [m]$
 - A is a distribution over the set of pairs (i, j) , where $1 \leq i \leq n$ and $1 \leq j \leq m$
 - ϵ
- **Output:**
 - Decide if the two coordinates of A are independent (or at least ϵ -close to being so)

Sublinearity

- A linear time algorithm for the problem may use $O(nm)$ samples

- We want a sublinear

Using this amount of samples, we can sample A until it is almost explicitly given – then testing for independence is trivial

as a

- We will show:
 - $O(n^{2/3}m^{1/3}\text{poly}(\log n, \epsilon^{-1}))$



More background

Now that we understand the issue

More background

- We will use a few more names for random variables:
 - Approximation
 - Restriction
 - Coarsening

Approximation

- Let A be a random variable over R
- Let $\varepsilon > 0$

- We say \tilde{A} Approximates A if:
 - For all i in R , $(1-\varepsilon)A(i) \leq \tilde{A} \leq (1+\varepsilon)A(i)$

Restriction

- Let X be distributed over the set R
- And let R' be a non-trivial sub-set of R
- X restricted over R' is the random variable $X_{|R'}$
 - For i in R' , $X_{|R'}(i) = X(i) / X(R')$
- Not the same is a conditioned variable
 - distributed over R' and not over R

Coarsening

- Let X be distributed over the set R
- Let R' be a partition of R
 - $R' = \{R_1, \dots, R_k\}$
- The coarsening of X over R' is the random variable $X_{\langle R' \rangle}$ over $[k]$
 - For i in $[k]$, $X_{\langle R' \rangle}(i) = X(R_i)$

Simple observation

- Let X be distributed over the set R
- Let R' be a partition of R
 - $R' = \{R_1, \dots, R_k\}$
- For all i in $[k]$ and j in R_i :
 - $X(j) = X_{\langle R' \rangle}(i) X_{|R_i}(j)$

$$X_{\langle R' \rangle}(i) = X(R_i)$$

$$X_{|R_i}(j) = X(j) / X(R_i)$$

Simple Observation

- Say A is distributed over $[n] \times [m]$
- Define $R' = \{R_1, \dots, R_n\}$, With:
 - $R_i = \{(i, j) \mid j \text{ in } [m]\}$
- It follows that:
 - $A(i, j) = (\pi_1 A)(i) \cdot (\pi_2 A_{\{i\} \times [m]})(j)$


$$X_{\langle R' \rangle}(i)$$


$$X_{|R_i}(j)$$

Closeness again

- Lemma:
 - Let X and Y be random variables, both distributed over the set R
 - Let $R' = \{R_1, \dots, R_k\}$ be a partition of R
 - If:
 - $|X_{\langle R' \rangle} - Y_{\langle R' \rangle}| \leq \varepsilon_1$
 - For all i in $[k]$, $|X_{|R_i} - Y_{|R_i}| \leq \varepsilon_2$
 - Then:
 - $|X - Y| \leq \varepsilon_1 + \varepsilon_2$

Closeness again

- Lemma:
 - Let X and Y be random variables, both distributed over the set R
 - Let R' be a non trivial sub-set of R
 - Then:
 - $|X_{|R'} - Y_{|R'}| \leq 2|X - Y| / X(R')$
- In words, if X and Y are close, then they are close when restricted to sufficiently 'heavy' partitions.



Tools

What we're about to use

Tools

- We'll review two tools for random variables manipulation:
 - Bucketing
 - Other tools (given 'for free')

Variable uniformity

- Given a black-box distributions X over R , there is a test that requires:
 - $O(\varepsilon^{-4}|R|^{1/2} \log|R| \log(1/\delta))$ samples
- Outputs 'PASS' with probability $1 - \delta$ if:
 - $X = U_R$
- Outputs 'FAIL' with probability $1 - \delta$ if:
 - $|X - U_R| \geq \varepsilon$

Bucketing - Motivation

- There is a test for independence, if the distribution's projections are uniform
 - Can only be independent if A is $U_{S \times T}$
- Wishful thinking: maybe some coarsening of A might achieve uniform distributions over each partition (or at least get close to it)

Bucketing

- Given an explicit random variable X , we want to partition its values into buckets
- When restricted to a bucket, X should be close to uniform

Bucketing

- We define Bucket(X, R, ε) as the following partition of R :
 - $\text{Bucket}(X, R, \varepsilon) = \{R_0, R_1, \dots, R_k\}$
- Where:
 - $k = (2/\log(1+\varepsilon)) \cdot \log(|R|)$
 - $R_0 = \{i \mid X(i) < 1/(|R| \log|R|)\}$
 - For all i in $[k]$:
 - $R_i = \left\{ j \mid \frac{(1+\varepsilon)^{i-1}}{|R| \log|R|} \leq X(j) < \frac{(1+\varepsilon)^i}{|R| \log|R|} \right\}$

Bucketing

- It follows that:
 - $X(R_0) \leq 1/\log|R|$ $R_0 = \{i \mid X(i) < 1/(|R| \log|R|)\}$
 - Since R_0 is a small set, we will ignore it later on – this is where we keep the junk values of X

- We need to show that: Proof – on blackboard
 - For i in $[k]$, $|X_{|R_i} - U_{R_i}| \leq \epsilon$
 - That is, every $X_{|R_i}$ is close to uniform

Bucketing

- Say \tilde{A} approximates A
- Then, $\text{Bucket}(\tilde{A}, R, \varepsilon)$ has similar properties to $\text{Bucket}(A, R, \varepsilon)$:
 - $k = O(\varepsilon^{-1} \log |R|)$
 - For all i in $[k]$, $|A_{|R_i} - U_{R_i}| \leq 3\varepsilon$
 - $A(R_0) \leq (1 + \varepsilon) / \log |R|$

Other tools

- A test for black-box variables closeness
- An algorithm for estimating large portions of a random variable

Variables closeness

- Given two black-box distributions X and Y , with $\|X\|_\infty \leq \|Y\|_\infty$, there is a test requiring:
 - $O((|R|^2 \|X\|_\infty \|Y\|_\infty \varepsilon^{-4} + |R|^{1/2} \|X\|_\infty \varepsilon^{-2}) \log(1/\delta))$ samples
- That

The sample complexity contains the L-infinity norm of X and Y . If there are no values with high probability, this can be a good thing

 - If X and Y are close in L-infinity norm, the test outputs PASS with probability at least $1 - \delta$
 - If $\|X - Y\|_\infty > \varepsilon$, it outputs FAIL with probability at least $1 - \delta$

Variables closeness

- Given two black-box distributions X and Y , over R , there is a test that requires:
 - $O(|R|^{2/3}\epsilon^{-4} \log|R| \log(1/\delta))$ samples
- Outputs 'PASS' with probability $1 - \delta$ if X and Y are close
- Outputs 'FAIL' with probability $1 - \delta$ if:
 - $|X - Y| \geq \epsilon$

Variable estimation

- Given a black box distribution X over R , a threshold t and an accuracy $\varepsilon > 0$, there is an algorithm that requires:
 - $O(t^{-1} \varepsilon^{-2} \log|R| \log(1/\delta))$ samples
- Outputs an estimate \tilde{X} of X , such that with probability at least $1-\delta$, for every i in R :
 - If $X(i) \geq t$, $(1-\varepsilon)X(i) \leq \tilde{X}(i) \leq (1+\varepsilon)X(i)$
- Also outputs the set R' on which the approximation is guaranteed:
 - $R' = \{ i \in R \mid X(i) \geq t \}$



Problem 1

Enough with the intro, lets get to it!

Problem 1

- Input:
 - Black-box distribution A , over $[n] \times [m]$
 - A is a distribution over the set of pairs (i, j) , where $1 \leq i \leq n$ and $1 \leq j \leq m$
 - ϵ
- Output:
 - Decide if the two coordinates of A are independent (or at least ϵ -close to being so)

Solution Plan

- Main goal – perform a reduction:
 - A's distribution is unknown...
- Using distribution manipulation (Bucketing, Sieves, ...) we can reduce the problem to that of the uniform case
- That problem is then already solved

Solution Plan

- In order to achieve the best sampling complexity, we will separate the samples from A into two sets:
 - Heavy prefixes
 - Light prefixes
- The heavy prefixes set will contain all values i in $[n]$ that have “heavy” probabilities

Solution plan

- We will then test the heavy prefixes for independence
- Then the light prefixes will be tested
- At last, we will assert that the results are consistent with both sets – heavy and light

Separating heavy & light

- A is distributed over $S \times T$
- Let $0 < \alpha < 1$ be a parameter to be determined later
- We define:
 - $S' = \left\{ i \in [n] \mid (\pi_1 A)(i) \geq n^{-\alpha} \right\}$
 - $S'' = \left\{ i \in [n] \mid (\pi_1 A)(i) \geq \frac{1}{2} n^{-\alpha} \right\}$

Separating heavy & light

- Using a total of $O(n^\alpha \varepsilon^{-2} \log n)$ samples, we can estimate $(\pi_1 A)(i)$ for i in S'' , to within $\varepsilon/75$ factor
 - Here, $t = n^{-\alpha}/2$, $\delta = 1/n$
 - Doing so generates A'_1 – estimates $\pi_1 A$
- We define $Q = \left\{ i \in S'' \mid A'_1(i) \geq \frac{2}{3} n^{-\alpha} \right\}$
- Then, $Q \supset S'$
- Also, Q does not contain any values of i , for which $(\pi_1 A)(i) \leq (n^{-\alpha})/2$

Separating heavy & light

- We call Q the heavy prefixes
- In case $(\pi_1 A)(Q) \leq \epsilon/2$, the heavy prefixes cannot contribute more than $\epsilon/2$ to A 's distance from independence
- Otherwise, we can easily simulate the distribution for $A_{|Q \times [m]}$
 - We sample A until we find a valid member, which we output – using $o(\epsilon^{-1} \log(nm))$ samples promises high probability for success in doing this

Separating heavy & light

- We call $[n] \setminus Q$ the light prefixes
- Again if $(\pi_1 A)([n] \setminus Q) \leq \epsilon/2$, it cannot contribute much to A 's distance from independence
- And again, we can simulate the distribution for $A_{|([n] \setminus Q) \times [m]}$



The heavy prefixes

The Heavy prefixes

- We give an algorithm that:
 - If A is independent, outputs 'PASS'
 - If A is not 3ε -independent, outputs 'FAIL' with high probability
- We will use sieves in order to generate a fully uniformed distribution

Sieves - Motivation

- As we saw earlier, bucketing partitions a distribution into close to uniform parts
- If we could only make those parts truly uniform, testing would be easy – we will need to test closeness against the uniform distribution over $S \times T$

Sieves

- A sieve is a function, that generates variable B from variable A
- The sieve samples A , and sifts its output in a way that achieves B 's distribution (say, uniform)

Sieves

- A sieve works in batch mode
 - Collects samples from A , and generates t samples from B
- For a given t samples wanted (batch size), the sieve will sample A
- We say the sample complexity of the sieve, is the maximum number of samples from A it uses, for a given t
 - A function of A 's domain as well as t

Sieves

- An (A, B) -sieve is specified in terms of relationship between B and A
- In our case:
 - A 's projections are close to uniform
 - B is close to A , and has uniform projections
 - B preserves A 's independence (if existed)

Sieves

- Lemma:
 - There exists an (A, B) -sieve for random variables over $S \times T$, such that for any t , with high probability:
 - If $|\pi_1 A - U_s| \leq \varepsilon/4$, then $|A - B| \leq \varepsilon$
 - If $A = (\pi_1 A) \times (\pi_2 A)$, then $B = (U_s) \times (\pi_2 A)$
 - The sieve's sample complexity is:
 - $O(\max\{|S|, t\} \log^3(\max\{|S|, t\}))$

Composing sieves

- Sieves can be composed. Since the input and output distributions are ϵ -close, properties will remain
- An (A, C) -sieve can be composed with a (C, B) -sieve:
 - We get an (A, B) -Sieve

Composing sieves

- The sampling complexity is also composed:
 - (A, C)-sieve sampling complexity is $f(t)$
 - (C, B)-sieve sampling complexity is $g(t)$
 - Composed (A, B)-sieve sampling complexity, $h(t)$, is:
 - t samples from B require $g(t)$ samples from C, using the (C, B)-sieve. That requires $f(g(t))$ samples from A, using the (A, C)-sieve
 - $h(t) = f(g(t))$

Composing sieves

- Lemma:
 - There exists an (A, B) -sieve for random variables over $S \times T$ such that if
 - $|\pi_1 A - U_S| \leq \varepsilon/25$
 - $|\pi_2 A - U_T| \leq \varepsilon/25$
 - Then, with high probability:
 - $|B - A| \leq 24\varepsilon/25$
 - If $A = (\pi_1 A) \times (\pi_2 A)$, then $B = U_{S \times T}$
 - If A is not ε -independent, then $|B - U_{S \times T}| \geq \varepsilon/25$
 - The sample complexity of the sieve is:
 - $O(\max\{|S| + |T|, t\} \log^6(\max\{|S|, |T|, t\}))$

The heavy prefixes

TestHeavyIndependence(A, A', ϵ)

- 1) $\hat{S} = \{S_0, S_1, \dots, S_k\} = \text{Bucket}(A', Q, \epsilon/75)$
- 2) Obtain A'_2 , which approximates $\pi_2 A$ to within $\epsilon/75$, on a set \check{T} that includes all j in $[m]$ with probability at least $(m \log m)^{-1}$
- 3) $\check{T} = \{T_0, T_1, \dots, T_l\} = \text{Bucket}(A'_2, \check{T}, \epsilon)$
 - 1) Add \check{T} to T_0
- 4) For (S_i, T_j) , i in $[k]$, j in $[l]$ do:
- 5) If $A(S_i \times T_j)$ is not small, then:
- 6) If $\pi_1 A_{|S_i \times T_j}$ or $\pi_2 A_{|S_i \times T_j}$ are not both $\epsilon/25$ -uniform, or if $A_{|S_i \times T_j}$ is not ϵ -independent, then FAIL
- 7) If $A_{\langle \hat{S} \times \check{T} \rangle}$ is not $\epsilon/2$ -independent then FAIL
- 8) PASS

Algorithm analysis

- Line 7 says:
 - If $A_{\langle \hat{S}, \mathbb{F} \rangle}$ is not $\varepsilon/2$ -independent then FAIL
- This can be tested using brute-force, since the variables are logarithmic in $|S|$ and $|T|$

Algorithm analysis

- Line 5 says:
 - If $A(S_i \times T_j)$ is not small, then:
- Some sets might not contribute much to A 's distance from independence
- We distinguish between two cases:
 - $A(S_i \times T_j) \geq \epsilon/(kl)$
 - $A(S_i \times T_j) \leq \epsilon/(2kl)$
- We do so by taking $\tilde{O}(kl/\epsilon)$ samples of A , and counting those in $S_i \times T_j$

Algorithm analysis

- Line 6 says:
 - If $\pi_1 A_{|S_i \times T_j}$ or $\pi_2 A_{|S_i \times T_j}$ are not both $\epsilon/25$ -uniform, or if $A_{|S_i \times T_j}$ is not ϵ -independent, then FAIL
- We can test that a variable is $\epsilon/25$ -uniform, using $O(\epsilon^{-4} n^{1/2})$ samples of $A_{|S_i \times T_j}$
- In order to sample $A_{|S_i \times T_j}$ we must repeatedly sample A , until a valid sample appears
 - Since step 5 assured $A(S_i \times T_j) \geq \epsilon/(2kl)$, we might need $O(\epsilon^{-1} \log^3(nm))$ samples of A to generate one sample of $S_i \times T_j$

Algorithm analysis

- All that's left is to test if $A_{|S_i \times T_j}$ is ε -independent
- Remember that from bucketing an approximation of A we get:
 - For all i in $[k]$, $|\pi_1 A - U_{S_i}| \leq 3(\varepsilon/75) = \varepsilon/25$
 - For all j in $[l]$, $|\pi_2 A - U_{T_j}| \leq 3(\varepsilon/75) = \varepsilon/25$

Testing independence

- We need an algorithm for the following problem:
 - A is a black-box distribution over $S \times T$
 - $|\pi_1 A - U_S| \leq \epsilon/25$, $|\pi_2 A - U_T| \leq \epsilon/25$
 - Decide if A is ϵ -independent, with high probability
- Note that A is either independent or ϵ -far from $U_{S \times T}$
 - An algorithm can PASS if independent, and FAIL if ϵ -far from $U_{S \times T}$

Composing sieves

- Lemma:
 - There exists an (A, B) -sieve for random variables over $S \times T$ such that if
 - $|\pi_1 A - U_S| \leq \varepsilon/25$
 - $|\pi_2 A - U_T| \leq \varepsilon/25$
 - Then, with high probability:
 - $|B - A| \leq 24\varepsilon/25$
 - If $A = (\pi_1 A) \times (\pi_2 A)$, then $B = U_{S \times T}$
 - If A is not ε -independent, then $|B - U_{S \times T}| \geq \varepsilon/25$
 - The sample complexity of the sieve is:
 - $O(\max\{|S| + |T|, t\} \log^6(\max\{|S|, |T|, t\}))$

Testing independence

- We can apply that sieve, and test B for uniformity (with accuracy greater than $\epsilon/25$)
- Testing B for uniformity will cost:
 - $\tilde{O}(\epsilon^{-1}|S \times T|^{1/2})$ samples from the sieve
- Total sampling complexity from A, for each pair of buckets:
 - $\tilde{O}(\epsilon^{-4}(|S|+|T|) \log^6(\epsilon^{-4}(|S|+|T|)))$
- This should be multiplied by the number of buckets used: $O(\epsilon^{-1} \log^3(nm))$

Algorithm analysis

- Back to the algorithm – we will now prove it works
- If A is independent, every restriction $A_{|S_i \times T_j}$ is independent
- Also, $A_{\langle \hat{S} \times \bar{T} \rangle}$ is independent
- Therefore, if A is independent it passes the test

Algorithm analysis

- Say A passes the test
- In step 6 we made sure the restricted distributions are $\varepsilon/25$ -uniform
- Passing the sieve test will require:
 - $|A_{|S_i \times T_j} - U_{S_i \times T_j}| \leq \varepsilon$
- Also, $A_{\langle \hat{S} \times \bar{T} \rangle}$ is $\varepsilon/2$ -independent
- There exists an independent distribution D over $[k] \times [l]$ such that:
 - $|A_{\langle \hat{S} \times \bar{T} \rangle} - D| \leq \varepsilon/2$

Algorithm analysis

- B is a new random variable, that randomly picks (i, j) from D
- Then it randomly picks (i', j') from S_i, T_j uniformly
- B is independent

Closeness again

- Lemma:
 - Let X and Y be random variables, both distributed over the set R
 - Let $R' = \{R_1, \dots, R_k\}$ be a partition of R
 - If:
 - $|X_{\langle R' \rangle} - Y_{\langle R' \rangle}| \leq \varepsilon_1$
 - For all i in $[k]$, $|X_{|R_i} - Y_{|R_i}| \leq \varepsilon_2$
 - Then:
 - $|X - Y| \leq \varepsilon_1 + \varepsilon_2$

Algorithm analysis

- According to the closeness lemma:
- $|A - B| \leq \varepsilon + \varepsilon/2 + \varepsilon + (1 - \varepsilon)/\log n \leq 3\varepsilon$
 - First and second terms are from the lemma
 - Third term comes from possibly skipping pairs of buckets with probability $< \varepsilon/(lk)$
 - Fourth term comes from ignoring $A(T_0)$

Final complexity

- The complexity is dominated by each pair of buckets being tested with the sieve
- Total sample complexity:
 - $\tilde{O}((|S|+|T|)\text{poly}(\varepsilon^{-1}))$ samples
 - If $|T|$ is small, this is almost linear – only good for large $|T|$
- With the heavy prefixes:
 - $|S| = O(n^\alpha)$
 - $|T| = O(m)$



The light prefixes

The light prefixes

- We give an algorithm that:
 - If A is independent, outputs 'PASS'
 - If A is not 3ε -independent, outputs 'FAIL' with high probability
- We use the L_1 distance test

The light prefixes

TestLightIndependence(A, ϵ)

- 1) Obtain an approximation A'_2 of $\pi_2 A$ within $\epsilon/75$ factor, on T' which includes all j in $[m]$ with probability at least $(m \log m)^{-1}$
- 2) $\mathcal{T} = \{T_0, T_1, \dots, T_l\} = \text{Bucket}(A'_2, T', \epsilon)$
 - 1) Add $T \setminus T'$ to T_0
- 3) For $j = 1, \dots, l$ do
- 4) If $A(S \times T_j)$ is not small then
- 5) If $|A_{|S \times T_j} - (\pi_1 A_{|S \times T_j}) \times (\pi_2 A_{|S \times T_j})| \geq \epsilon$, then FAIL
- 6) Let j' be such that $A(S \times T_{j'}) \geq \epsilon/(4l)$
- 7) For $j = 1, \dots, l$
- 8) If $A(S \times T_j)$ is not small then
- 9) If $|\pi_1 A_{|S \times T_j} - \pi_1 A_{|S \times T_{j'}}| \geq \epsilon$, then FAIL
- 10) PASS

Algorithm analysis

- If A is indeed independent, it passes
- If A passes the test, then:
 - Each of \mathcal{F} 's buckets (sufficiently large) are indeed ε -independent
 - $\pi_{I_j} A_{|S \times T_j}$ are all ε -close to some $\pi_{I_j} A_{|S \times T_j}$,
- Therefore, A is at least 3ε -independent

Algorithm analysis

- Lines 4 and 8 say:
 - If $A(S \times T_j)$ is not small then
- Similarly to before, we say $A(S \times T_j)$ is small, if:
 - $A(S \times T_j) \leq \varepsilon/(4l)$
- We do so by taking $\tilde{O}(l/\varepsilon)$ samples of A , and counting those in $S \times T_j$
 - Guarantees hitting a sample from $S \times T_j$ with $O(\text{poly}(\log(nm))l/\varepsilon)$ samples of A in lines 5,9

Algorithm analysis

- Line 6 says:
 - Let j' be such that $A(S \times T_{j'}) \geq \epsilon/(4l)$
- This can be achieved with choosing j' which passed as 'not small', in line 4
 - If none existed – no need to choose one

Algorithm analysis

- Line 5 says:
 - If $|A_{|S \times T_j} - (\pi_1 A_{|S \times T_j}) \times (\pi_2 A_{|S \times T_j})| \geq \epsilon$, then FAIL
- The projections of $A_{|S \times T_j}$ are sampled by sampling $A_{|S \times T_j}$, and ignoring a coordinate
- The test used for distance between distributions (also used in line 9)...

Variables closeness

- Given two black-box distributions X and Y , with $\|X\|_\infty \leq \|Y\|_\infty$, there is a test requiring:
 - $O((|R|^2 \|X\|_\infty \|Y\|_\infty \varepsilon^{-4} + |R|^{1/2} \|X\|_\infty \varepsilon^{-2}) \log(1/\delta))$ samples
- That:
 - If X and Y are close, it outputs PASS with probability at least $1 - \delta$
 - If $|X - Y| > \varepsilon$, it outputs FAIL with probability at least $1 - \delta$

Variables closeness

- Given two black-box distributions X and Y , over R , there is a test that requires:
 - $O(|R|^{2/3}\epsilon^{-4} \log|R| \log(1/\delta))$ samples
- Outputs 'PASS' with probability $1 - \delta$ if:
 - X and Y are close
- Outputs 'FAIL' with probability $1 - \delta$ if:
 - $|X - Y| \geq \epsilon$

Algorithm analysis

- We will use the first test for line 5:
 - If $|A_{|S \times T_j} - (\pi_1 A_{|S \times T_j}) \times (\pi_2 A_{|S \times T_j})| \geq \varepsilon$, then FAIL
- We will use the second test, for line 9:
 - If $|\pi_1 A_{|S \times T_j} - \pi_1 A_{|S \times T_j}| \geq \varepsilon$, then FAIL

Complexity analysis

- Complexity for steps 6 and 7-9 is dominated by that of steps 3-5
- Sample complexity for steps 7-9 (all included):
 - $\tilde{O}(|S|^{2/3}\epsilon^{-5})$

Complexity analysis

- We must bound the $\| \cdot \|_\infty$ part, for complexity analysis
- Because $\| \pi_1 A \|_\infty$ is bounded with the light prefixes, we get that:
 - $\| \pi_1 A_{|S \times T_j} \|_\infty \leq (2|S|^{-\alpha})/\varepsilon$, for every T_j
- Because of the bucketing:
 - $\| \pi_2 A_{|S \times T_j} \|_\infty \leq (1+3\varepsilon)|T_j|^{-1}$

Complexity analysis

- Sample complexity for steps 3-5 is given by $\log|T|$ times the sample complexity for each iteration j
- That is given by the theorem:
 - $\tilde{O}((1+3\varepsilon)(|S||T_j|)^2|S|^{-\alpha}(|S|^{-\alpha}|T_j|^{-1})\varepsilon^{-5})$
- That, times the sampling cost for sampling from restrictions:
 - $\tilde{O}(l/\varepsilon)$

Final complexity

- Putting it all together, we get:
 - $\tilde{O}((|S|^{2/3} + |S|^{2-2\alpha}|T|)\text{poly}(\varepsilon^{-1}))$
 - $|S| = O(n)$
 - $|T| = O(m)$
- If $|T|$ is really big, say $|T| = |S|$, then this complexity can be linear (if α is small)



Final algorithm

Putting the heavy and light prefixes back together

Putting them together

- We will now describe an algorithm for the general case
- As planned, it will first split values to heavy and light
- Will output PASS if A is independent
- Will output FAIL with high probability, if A is not 4ϵ -independent

Putting them together

TestIndependence(A, n, m, ϵ)

- 1) Let β be such that $m = n^\beta$.
Set $\alpha = (2+\beta)/3$
- 2) Obtain an approximation A'_1 of $\pi_1 A$ to within an $\epsilon/75$ factor, on S' which includes all i in $[n]$ which have probability at least $n^{-\alpha}$, and no i in $[n]$ with probability at most $n^{-\alpha}/2$
- 3) If $(\pi_1 A)(S')$ is not small, then:
 - 4) If TestHeavyIndependence($A_{|S' \times [m]}, A'_1_{|S' \times [m]}, \epsilon$) fails, then FAIL
- 5) If $(\pi_1 A)([n] \setminus S')$ is not small, then:
 - 6) If TestLightIndependence($A_{|[n] \setminus S' \times [m]}, \epsilon$) fails, then FAIL
- 7) If both $(\pi_1 A)(S')$ and $(\pi_1 A)([n] \setminus S')$ are not small, then:
 - 8) if $\pi_2 A_{|S' \times [m]}$ and $\pi_2 A_{|[n] \setminus S' \times [m]}$ are not ϵ -close, then FAIL
- 9) PASS

Algorithm analysis

- Steps 3, 5 and 7 say:
 - If ... is not small then:
- We say S' is small if:
 - $|(\pi_1 A)(S')| \leq \epsilon/2$

Algorithm analysis

- If A is independent, then the algorithm outputs PASS, with high probability
- Say the algorithm outputs PASS
- We partitioned A into two sets:
 - $Q = \{S' \times [m], ([n] \setminus S') \times [m]\}$
- As we saw earlier Q is a coarsening of A

Algorithm analysis

- Line 8 guarantees that the coarsening Q is not ε -far from independent
- Because each set in the coarsening is not more than 3ε -independent, A is no more than 4ε -independent
- In case steps 4 or 6 are not performed, A is no more than 4ε -independent

Complexity analysis

- Because $|T|$'s size creates a trade-off between heavy and light prefixes, choosing α can create an optimal final complexity:
 - We chose $\alpha = (2+\beta)/3$
 - $\beta = \log m / \log n = \log_n m$
 - $n^\alpha = n^{2/3} n^{\beta/3} = n^{2/3} m^{1/3}$

Complexity analysis

- Heavy complexity:
 - $\tilde{O}((|S|+|T|)\text{poly}(\varepsilon^{-1}))$
 - $|S| = O(n^\alpha)$, $|T| = O(m)$
- We get: $\tilde{O}((n^\alpha+m)\text{poly}(\varepsilon^{-1}))$
- Since $n^\alpha = n^{2/3}m^{1/3}$, we finally get:
 - $\tilde{O}(n^{2/3}m^{1/3}\text{poly}(\varepsilon^{-1}))$

Complexity analysis

- Light complexity:
 - $\tilde{O}((|S|^{2/3} + |S|^{2-2\alpha}|T|)\text{poly}(\varepsilon^{-1}))$
 - $|S| = O(n)$, $|T| = O(m)$
- We get: $\tilde{O}((n^{2/3} + n^{2-2\alpha}m)\text{poly}(\varepsilon^{-1}))$
- Since $n^\alpha = n^{2/3}m^{1/3}$, we finally get:
 - $\tilde{O}(n^{2/3}m^{1/3}\text{poly}(\varepsilon^{-1}))$

Final complexity

- The algorithm's complexity is dominated by the heavy and light tests
- Both tests have the same complexity:
 - $\tilde{O}(n^{2/3}m^{1/3} \text{poly}(\epsilon^{-1}))$

Summary

- We reduced the problem of independence testing to that of the uniform case:
 - By using buckets we achieved uniform closeness
 - By using sieves we achieved fully uniform distributions
- After that, we tested against an independent uniform distribution
- Made sure the results are consistent

Homework ☺

- show that if $|A - B| \leq \epsilon/3$, and B is independent, Than:
 - $|A - (\pi_1 A) \times (\pi_2 A)| \leq \epsilon$
- Prove that if $|X_{\langle R_i \rangle} - Y_{\langle R_i \rangle}| \leq \epsilon_1$ and for all i in $[k]$, $|X_{|R_i} - Y_{|R_i}| \leq \epsilon_2$, then:
 - $|X - Y| \leq \epsilon_1 + \epsilon_2$
- Prove that bucketing an approximation keeps the properties of normal bucketing (say \tilde{A} approximates A, and look at $\text{Bucket}(\tilde{A}, \epsilon, R)$):
 - $k = O(\epsilon^{-1} \log |R|)$
 - For all i in $[k]$, $|A_{|R_i} - U_{R_i}| \leq 3\epsilon$
 - $A(R_0) \leq (1+\epsilon) / \log |R|$