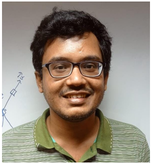
Towards Practical Distribution Testing

Yash Pote, National University of Singapore

Kuldeep S. Meel

Gunjan Kumar



Sayantan Sen

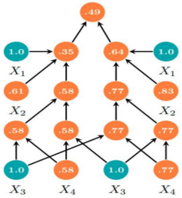
R. Bhattacharyya

S. Chakraborty

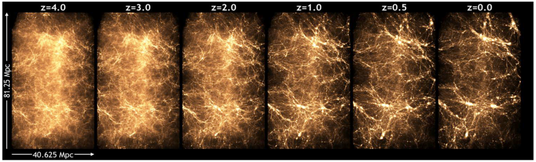
Uddalok Sarkar

High-Dimensional Distributions in Practice

- model data with complex relationships
  - text
  - images
  - signals
  - molecular structure



- high-dimensional sampling is a key part of
  - software testing
  - scientific simulations
  - generative modeling



High-Dimensional Distributions: Key Challenges

Sampling from high-dimensional distributions:

- ☹ hard for most real-world problems
  - ◦ subgraph sampling
  - ◦ sampling strings from grammars

- easy to get wrong
  - ◦ developers use ad-hoc methods
    - ◦ for e.g., most MCMC is run with heuristically chosen parameters [G04]
  - ◦ failures are silent

- a question every developer has: is my algorithm correct?
  - ◦ is the output distribution $\mathcal{Q}$ of an algorithm close to the intended target $\mathcal{P}$?

High-Dimensional Distributions: Key Challenges

Sampling from high-dimensional distributions:

- ☹ hard for most real-world problems
  - subgraph sampling
  - sampling strings from grammars

- easy to get wrong
  - developers use ad-hoc methods
    - for e.g., most MCMC is run with heuristically chosen parameters [G04]
  - failures are silent

- a question every developer has: is my algorithm correct?
  - is the output distribution $\mathcal{Q}$ of an algorithm close to the intended target $\mathcal{P}$?

Goals

Given distance metric $d(\cdot, \cdot)$, and parameters $\varepsilon < \eta$, determine whether distributions $\mathcal{P}$ and $\mathcal{Q}$ are close:

(Goal 1) Decision: $d(\mathcal{P}, \mathcal{Q}) < \varepsilon$ v.s. $d(\mathcal{P}, \mathcal{Q}) > \eta$

(Goal 2) Estimation: $d(\mathcal{P}, \mathcal{Q})$ upto $\pm\varepsilon$ (In focus today)

Outline

- ☐ A motivating instance
- ☐ Formalizing the problem
- ☐ The estimation problem
- ☐ Ongoing work

Outline

- ☐ A motivating instance
- ☐ Formalizing the problem
- ☐ The estimation problem
- ☐ Ongoing work

Motivating Instance

A developer writes a buggy program SHUFFLE-1 to shuffle an array `A` of length `n`. How can we detect the bug?

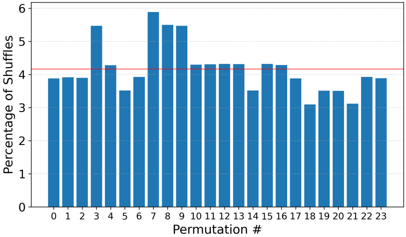
```
SHUFFLE-1(A,n)

for j in 1,2,...,n
   r ∼ sample([1,2,...,n])
   swap A[r] and A[j]
return A
```

```
FISHER-YATES(A,n)

for j in 1,2,...,n
  r ∼ sample([j,j+1,...,n])
  swap A[r] and A[j]
return A
```

Approach 1 - Blackbox Access

A developer writes a buggy program SHUFFLE-1 to shuffle an array `A` of length `n`. How can we detect the bug?

```
SHUFFLE-1(A,n)

for j in 1,2,...,n
  r ~ sample([1,2,...,n])
  swap A[r] and A[j]
return A
```



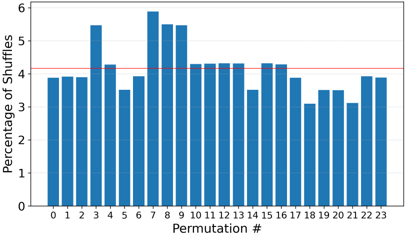
The blackbox approach:

- method - execute program multiple times, outputs are samples from distribution
  - if program correct - uniform distribution on permutations of A
  - if program not correct - output far from uniform
- what we observe
  - for $n = 4$, $d_{TV}(\text{Uniform}, \texttt{Shuffle-1}) \approx 0.06$, i.e. far from uniform

Approach 1 - Blackbox Access

A developer writes a buggy program SHUFFLE-1 to shuffle an array `A` of length `n`.
How can we detect the bug?

```
SHUFFLE-1(A,n)

for j in 1,2,...,n
  r ∼ sample([1,2,...,n])
  swap A[r] and A[j]
return A
```



The blackbox approach:

- method - execute program multiple times, outputs are samples from distribution
  - if program correct - uniform distribution on permutations of A
  - if program not correct - output far from uniform
- what we observe
  - for $n = 4$, $d_{TV}(\text{Uniform}, \texttt{Shuffle-1}) \approx 0.06$, i.e. far from uniform

Drawback:

- high sample complexity: at least $\sqrt{n!} = 2^{\Omega(n)}$ samples needed

Approach 2 - Whitebox Access

A developer writes a program SHUFFLE-1 to shuffle an array `A` of length `n`.
How can we detect the bug?

```
SHUFFLE-1(A,n)

for j in 1,2,...,n
   r ~ sample([1,2,...,n])
   swap A[r] and A[j]
return A
```

```
FISHER-YATES(A,n)

for j in 1,2,...,n
   r ~ sample([j,j+1,...,n])
   swap A[r] and A[j]
return A
```

The whitebox approach:

- method - formal analysis of the code
- observation - bug because the #of shuffles($n!$) does not divide #executions(n^n)

Approach 2 - Whitebox Access

A developer writes a program SHUFFLE-1 to shuffle an array `A` of length `n`.
How can we detect the bug?

```
SHUFFLE-1(A,n)

for j in 1,2,...,n
   r ~ sample([1,2,...,n])
   swap A[r] and A[j]
return A
```

```
FISHER-YATES(A,n)

for j in 1,2,...,n
   r ~ sample([j,j+1,...,n])
   swap A[r] and A[j]
return A
```

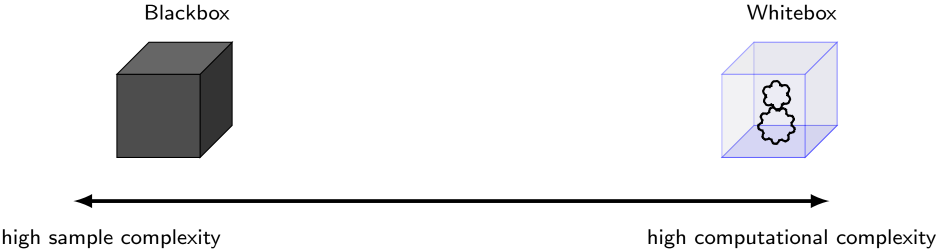
The whitebox approach:

- method - formal analysis of the code
- observation - bug because the #of shuffles($n!$) does not divide #executions(n^n)

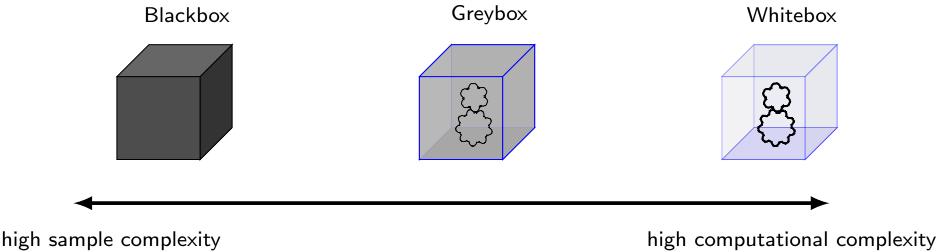
Drawback:

- computationally intractable: for high-dimensional ($n >> 1$) programs, method is intractable
  - #P-hard to count #executions

Approach 3 - Greybox Access



Approach 3 - Greybox Access



Greybox approach (Our Contribution)

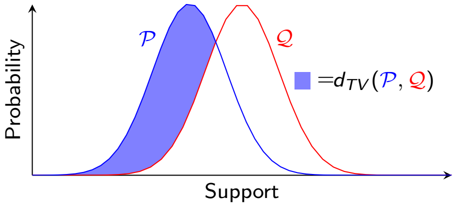
- ☺ more powerful, yet computationally feasible queries
- ☺ rich set of queries that adapt to underlying distributions
- ☺ scales to high-dimensional distributions

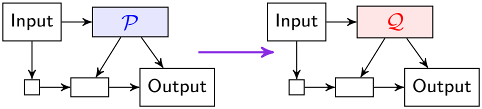
Outline

- ☑ A motivating instance
- ☐ Formalizing the problem
- ☐ The estimation problem
- ☐ Ongoing work

Distance - Total Variation

$$d_{TV}(\mathcal{P}, \mathcal{Q}) = \sum_{\sigma \in \{0,1\}^n} \max(\mathcal{P}(\sigma) - \mathcal{Q}(\sigma), 0)$$





- Fundamental metric across computer science
  - Approximation algo.
  - Learning

- Operational meaning:
  - program $\mathcal{A}$ uses a sample from $\mathcal{P}$
  - output of $\mathcal{A}$ is a distribution
  - if we replace $\mathcal{P}$ with $\mathcal{Q}$, the probability of Bad event in the output of $\mathcal{A}$ increases at most by $d_{TV}(\mathcal{P}, \mathcal{Q})$

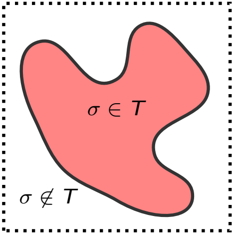
Conditioning: The Magic Ingredient behind Graybox Approach

Conditioning access

Given a distribution $\mathcal{Q}$, we can

- Specify a set $T \subseteq \{0,1\}^n$
- Draw samples according to the distribution $\mathcal{Q}_T$, i.e.

$$\mathcal{Q}_T(\sigma) = \frac{\mathcal{Q}(\sigma)}{\sum_{\sigma \in T} \mathcal{Q}(\sigma)}$$



Conditioning: The Magic Ingredient behind Graybox Approach

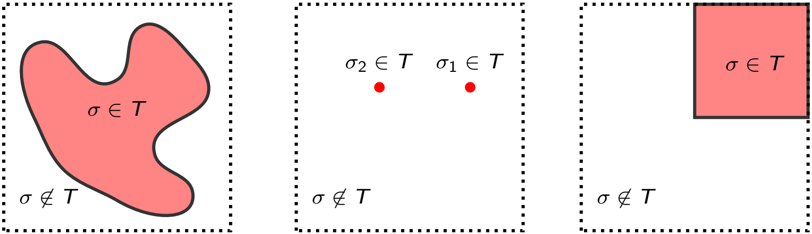
Conditioning access

Given a distribution $\mathcal{Q}$, we can

- Specify a set $T \subseteq \{0,1\}^n$
- Draw samples according to the distribution $\mathcal{Q}_T$, i.e.

$$\mathcal{Q}_T(\sigma) = \frac{\mathcal{Q}(\sigma)}{\sum_{\sigma \in T} \mathcal{Q}(\sigma)}$$

- Arbitrary sets T, well studied but unrealistic. Requires 2^n bits to specify.
- Restricted conditional variants of T *can* be implemented.
  - Pair : $|T| = 2$
  - Subcube : $T = 10 * * \triangleq \{1000, 1001, 1010, 1011\}$



Problem Statement - Distance Estimation

- estimation: How far is $\mathcal{P}$ from $\mathcal{Q}$?
- input:
  - distributions $\mathcal{P}$,$\mathcal{Q}$
  - tolerance: $0 < \varepsilon < 1$
- output:
  - return α s.t. $\Pr\left[\alpha \in d_{TV}(\mathcal{P}, \mathcal{Q}) \pm \varepsilon\right] > 2/3$
- query model:
  - conditional sampling

Outline

- ☑ A motivating instance
- ☑ Formalizing the problem
- ☐ Attacking the estimation problem
- ☐ Ongoing work

Background

Agenda

Design practical algorithms to take in a pair of distributions $\mathcal{P}$ and $\mathcal{Q}$, and a tolerance parameter ε, return $\alpha \in d_{TV}(\mathcal{P}, \mathcal{Q}) \pm \varepsilon$

Background

Agenda

Design practical algorithms to take in a pair of distributions $\mathcal{P}$ and $\mathcal{Q}$, and a tolerance parameter ε, return $\alpha \in d_{TV}(\mathcal{P}, \mathcal{Q}) \pm \varepsilon$

What was known?

- $\Theta(2^n/n)$ samples in the blackbox sampling model [VV13,17]
- Non-tolerant equivalence testing is in $\Theta(\log(n))$ in COND [CRS15, CFGM16,CCK23]
- *polylog*(n) query equivalence tests in restricted variants of COND [BC18, N21, BCSV23]
- $\Omega(\log(n))$ lower bound for COND
- $\Omega(n/\log(n))$ lower bound for SUBCOND

What was open?

- no *poly*(n) query algorithm for TV distance estimation in COND

Our contribution

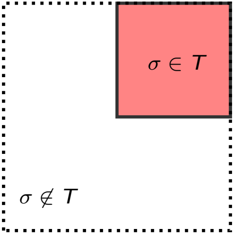
- first polynomial query distance estimator in COND
- $\mathcal{O}(n^3/\varepsilon^5)$ queries (equivalence) [KMP]
- $\mathcal{O}(n^2/\varepsilon^4)$ queries (identity) [BCPSS]

Subcube Conditioning

Description

Given a distribution $\mathcal{Q}$, we draw samples according to $\mathcal{Q}_T(\sigma) = \frac{\mathcal{Q}(\sigma)}{\sum_{\sigma \in T} \mathcal{Q}(\sigma)}$

- T is *subcube*, i.e. set of points with first m dimensions fixed to a string ρ
- Formally, $m \leq n$, $\rho \in \{0,1\}^{m \leq n}$, $T := \{\sigma | \sigma[1,m] = \rho\}$
- For example, $\rho = 10$, yields $T = 10 * * \triangleq \{1000, 1001, 1010, 1011\}$
- A natural model for
    - databases, cryptography, and approximate sampling



The Distance Estimator

INPUT:

- Distributions $\mathcal{P}$, $\mathcal{Q}$
- Tolerance ε

OUTPUT:

- α, such that $\Pr[d_{TV}(\mathcal{P}, \mathcal{Q}) - \varepsilon \leq \alpha \leq d_{TV}(\mathcal{P}, \mathcal{Q}) + \varepsilon] \geq 2/3$

The Distance Estimator

INPUT:

- Distributions $\mathcal{P}$, $\mathcal{Q}$
- Tolerance ε

OUTPUT:

- α, such that $\Pr[d_{TV}(\mathcal{P}, \mathcal{Q}) - \varepsilon \leq \alpha \leq d_{TV}(\mathcal{P}, \mathcal{Q}) + \varepsilon] \geq 2/3$

ALGORITHM:

Step 1: Draw samples $(\sigma_1, \sigma_2, \ldots \sigma_m) \sim \mathcal{P}$, where $m = \Theta\left(\frac{1}{\varepsilon^2}\right)$

Step 2: Find $\mathcal{Q}(\sigma_i)$

Step 3: Find $\mathcal{P}(\sigma_i)$

Step 4: Return $\frac{1}{m} \sum_{i \in [m]} \max\left(0, 1 - \mathcal{Q}(\sigma_i)/\mathcal{P}(\sigma_i)\right)$

The Distance Estimator

INPUT:

- Distributions $\mathcal{P}$, $\mathcal{Q}$
- Tolerance ε

OUTPUT:

- α, such that $\Pr[d_{TV}(\mathcal{P}, \mathcal{Q}) - \varepsilon \leq \alpha \leq d_{TV}(\mathcal{P}, \mathcal{Q}) + \varepsilon] \geq 2/3$

ALGORITHM:

Step 1: Draw samples $(\sigma_1, \sigma_2, \ldots \sigma_m) \sim \mathcal{P}$, where $m = \Theta\left(\frac{1}{\varepsilon^2}\right)$

Step 2: Find $\mathcal{Q}(\sigma_i)$ [no algo. known]

Step 3: Find $\mathcal{P}(\sigma_i)$ [$O(n^5)$ query algo. [CR14]]

Step 4: Return $\frac{1}{m} \sum_{i \in [m]} \max\left(0, 1 - \mathcal{Q}(\sigma_i)/\mathcal{P}(\sigma_i)\right)$

Agenda

Estimate $\mathcal{Q}(\sigma)$ and $\mathcal{P}(\sigma)$ using subcube conditional samples, faster than $O(n^5)$

The Distance Estimator: Estimating $\mathcal{P}(\sigma)$ in $O(n^2)$

INPUT:

- Distribution $\mathcal{P}$
- Tolerance ε
- Sample $\sigma \sim \mathcal{P}$

OUTPUT: $\frac{\mathcal{P}(\sigma)}{1+\varepsilon} \leq p \leq \mathcal{P}(\sigma)(1+\varepsilon)$

1 Let $k = 4n/\varepsilon^2$

2 For $0 \leq i \leq n-1$ [Stopping Rule, DKLR'00]

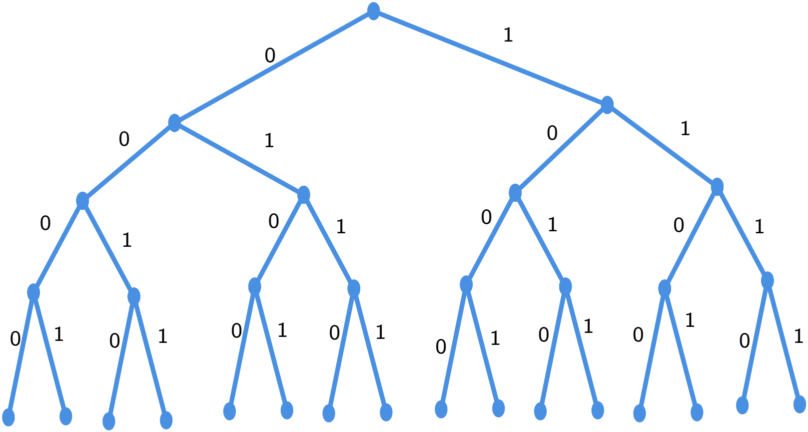
  a. conditioning on subcube $\sigma[1,i]$, sample from $\mathcal{P}_{\sigma[1,i]}$ until you see $\sigma[i+1]$ at least k times

  b. $r_i = \#$ of samples drawn in step (a)

3 Return $p = \prod_{i=0}^{n-1} k/r_i$

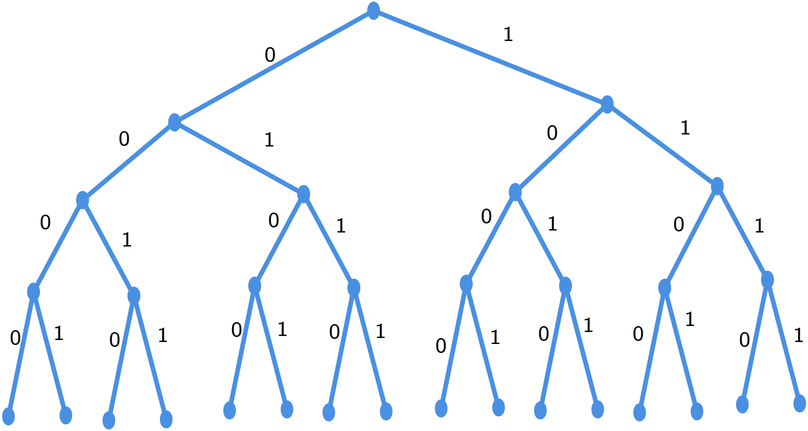
The Distance Estimator: Visualized

- Distribution $\mathcal{P}$ over $\{0,1\}^4$, and a sample 1001 sampled from $\mathcal{P}$



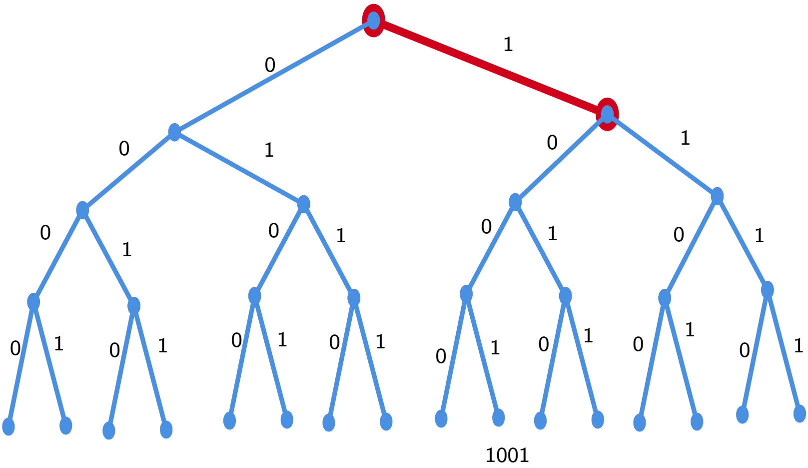
The Distance Estimator: Visualized

- Distribution $\mathcal{P}$ over $\{0,1\}^4$, and a sample 1001 sampled from $\mathcal{P}$
- Let $\mathcal{P}(10|1)$: Probability of 10, conditioned on 1.
  Chain rule: $\mathcal{P}(1001) = \mathcal{P}(1|\cdot) \times \mathcal{P}(10|1) \times \mathcal{P}(100|10) \times \mathcal{P}(1001|100)$



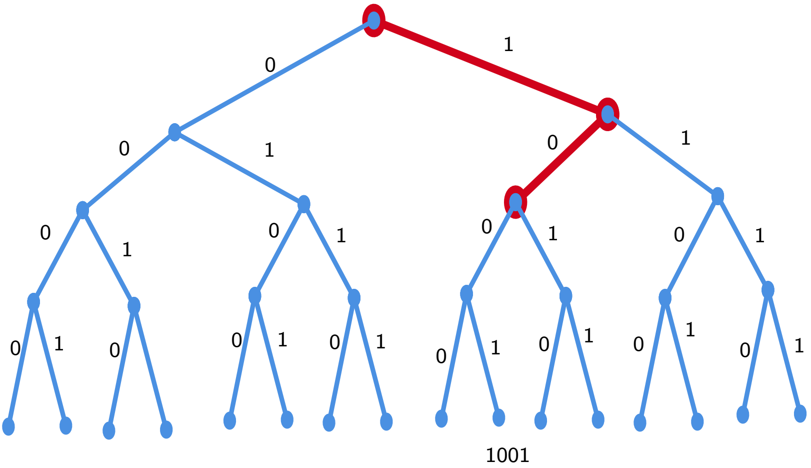
The Distance Estimator: Visualized

- Distribution $\mathcal{P}$ over $\{0,1\}^4$, and a sample 1001 sampled from $\mathcal{P}$
- Let $\mathcal{P}(10|1)$: Probability of 10, conditioned on 1.
  Chain rule: $\mathcal{P}(1001) = \boxed{\mathcal{P}(1|\cdot)} \times \mathcal{P}(10|1) \times \mathcal{P}(100|10) \times \mathcal{P}(1001|100)$



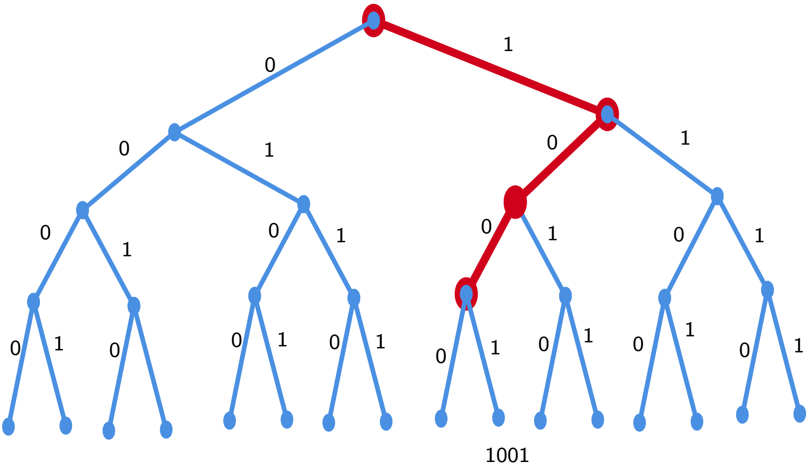
The Distance Estimator: Visualized

- Distribution $\mathcal{P}$ over $\{0,1\}^4$, and a sample 1001 sampled from $\mathcal{P}$
- Let $\mathcal{P}(10|1)$: Probability of 10, conditioned on 1.
  Chain rule: $\mathcal{P}(1001) = \boxed{\mathcal{P}(1|\cdot) \times \mathcal{P}(10|1)} \times \mathcal{P}(100|10) \times \mathcal{P}(1001|100)$



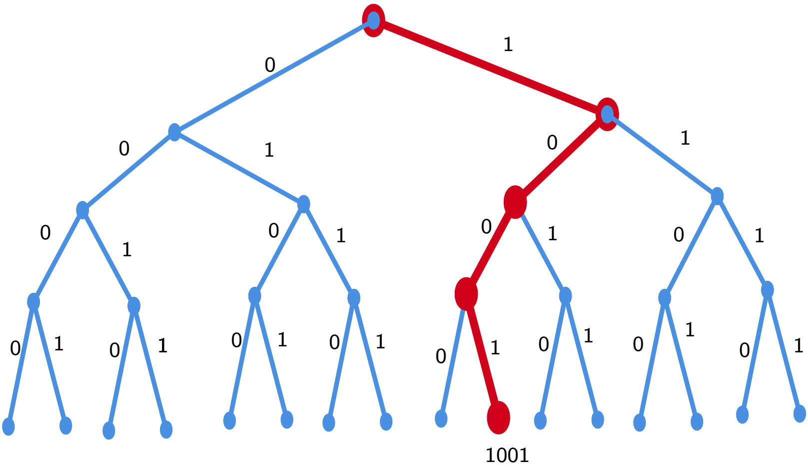
The Distance Estimator: Visualized

- Distribution $\mathcal{P}$ over $\{0,1\}^4$, and a sample 1001 sampled from $\mathcal{P}$
- Let $\mathcal{P}(10|1)$: Probability of 10, conditioned on 1.
  Chain rule: $\mathcal{P}(1001) = \boxed{\mathcal{P}(1|\cdot) \times \mathcal{P}(10|1) \times \mathcal{P}(100|10)} \times \mathcal{P}(1001|100)$



The Distance Estimator: Visualized

- Distribution $\mathcal{P}$ over $\{0,1\}^4$, and a sample 1001 sampled from $\mathcal{P}$
- Let $\mathcal{P}(10|1)$: Probability of 10, conditioned on 1.
  Chain rule: $\mathcal{P}(1001) = \boxed{\mathcal{P}(1|\cdot) \times \mathcal{P}(10|1) \times \mathcal{P}(100|10) \times \mathcal{P}(1001|100)}$



Challenges

- Challenge 1
  - estimating each marginal $\mathcal{P}(x|y)$ (upto an O(1) factor) requires $\propto \frac{1}{\mathcal{P}(x|y)}$ queries, where $\mathcal{P}(x|y)$ can be arbitrarily small,
  - there are n of these marginals, total complexity –

$$\sum^{n} \frac{1}{\mathcal{P}(x|y)}$$

Challenges

- Challenge 1
  - estimating each marginal $\mathcal{P}(x|y)$ (upto an O(1) factor) requires $\propto \frac{1}{\mathcal{P}(x|y)}$ queries, where $\mathcal{P}(x|y)$ can be arbitrarily small,
  - there are n of these marginals, total complexity –

$$\sum^{n} \frac{1}{\mathcal{P}(x|y)}$$

- Solution :- we prove $\sigma \sim \mathcal{P} \implies$ *in expectation all marginals are large i.e.,* $\Theta(1)$

$\boxed{\text{Complexity } \mathcal{O}(n^2)}$

Challenges

- Challenge 1
  - estimating each marginal $\mathcal{P}(x|y)$ (upto an O(1) factor) requires $\propto \frac{1}{\mathcal{P}(x|y)}$ queries, where $\mathcal{P}(x|y)$ can be arbitrarily small,
  - there are n of these marginals, total complexity –

$$\sum^{n} \frac{1}{\mathcal{P}(x|y)}$$

- Solution :- we prove $\sigma \sim \mathcal{P} \implies$ *in expectation all marginals are large i.e.,* $\Theta(1)$

$$\boxed{\text{Complexity } \mathcal{O}(n^2)}$$

- Challenge 2
  - sample $\sigma \sim \mathcal{P}$, estimate $\mathcal{Q}(\sigma)$
  - σ can have arbitrarily small marginals in $\mathcal{Q}$, even in expectation

Challenges

- Challenge 1
  - estimating each marginal $\mathcal{P}(x|y)$ (upto an O(1) factor) requires $\propto \frac{1}{\mathcal{P}(x|y)}$ queries, where $\mathcal{P}(x|y)$ can be arbitrarily small,
  - there are n of these marginals, total complexity –

$$\sum^{n} \frac{1}{\mathcal{P}(x|y)}$$

- Solution :- we prove $\sigma \sim \mathcal{P} \implies$ *in expectation all marginals are large i.e.,* $\Theta(1)$

$\boxed{\text{Complexity } \mathcal{O}(n^2)}$

- Challenge 2
  - sample $\sigma \sim \mathcal{P}$, estimate $\mathcal{Q}(\sigma)$
  - σ can have arbitrarily small marginals in $\mathcal{Q}$, even in expectation
- Solution :- we show that for every $\mathcal{D}$ there exists a distribution $\mathcal{D}'$ such that
  - $d_{TV}(\mathcal{D}, \mathcal{D}') \leq \varepsilon/10$
  - all marginals are large, i.e. $\mathcal{D}'(y|x) \in \Omega(\varepsilon/n)$
  - $\mathcal{D}'$ is easy to construct and sample from
- Finally - estimate $d_{TV}(\mathcal{P}, \mathcal{Q}')$ upto $\pm 8\varepsilon/10$ $\boxed{\text{Complexity } \mathcal{O}(n^3)}$

Open Questions

Ongoing work

Q1 What is the query complexity of estimating TV distance?

- In SUBCOND : $\mathcal{O}(n^3)$, $\Omega(n/\log(n))$ bound
- In COND : $\mathcal{O}(n^3)$, $\Omega(\log(n))$ bound

The road to practicality

Q2 Scaling our estimator to verify properties of large distributions

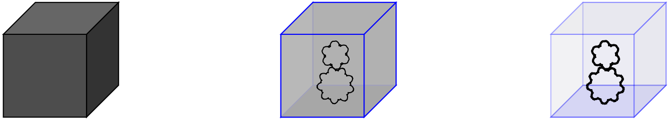
- How far is a quantized model from the original?
- How far is a current checkpoint from the desired distribution?

Practical Implications

- Tests are in use to verify correctness of combinatorial samplers
- Bugs discovered in several samplers that were in use [CM, MPC, PM, BCCMSS,KMP]
- Insights used for designing better sampling algorithms [GSCM, SGCM]

Conclusion

- Graybox sampling strikes a happy medium between computational intractable whitebox sampling, and statistically intractable blackbox

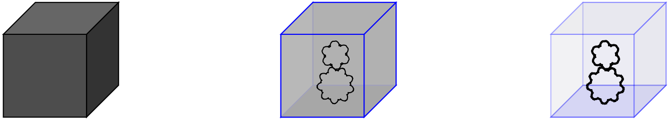


- Main takeaway:
  - theoretical insights useful in practical testing problems.
  - rich collection of practice inspired models left to be explored.
  - tight analysis, constants and instance optimality critical in practice

- Check our tools out at: `https://github.com/meelgroup/barbarik`

Conclusion

- Graybox sampling strikes a happy medium between computational intractable whitebox sampling, and statistically intractable blackbox



- Main takeaway:
  - theoretical insights useful in practical testing problems.
  - rich collection of practice inspired models left to be explored.
  - tight analysis, constants and instance optimality critical in practice
- Check our tools out at: `https://github.com/meelgroup/barbarik`

THANK YOU!