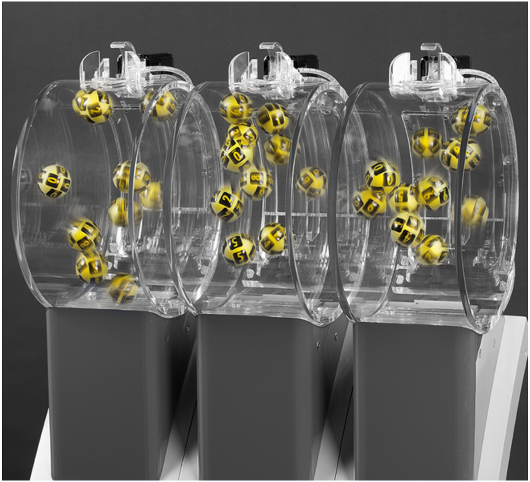
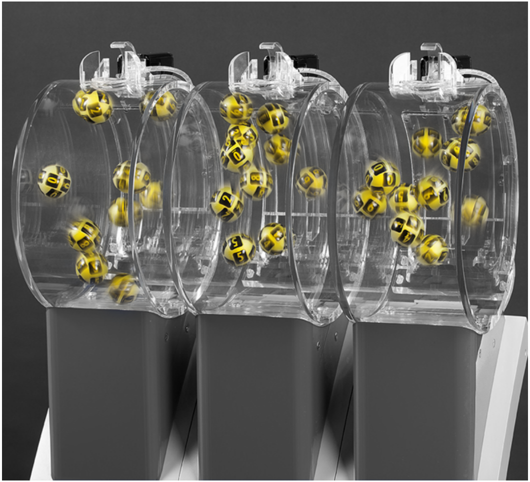
Property Testing:
Dealing with one aspect of Big Data

Sourav Chakraborty

A teaser: What is this?



Guiding Problem



How can you test if the machine is truely random?

Property Testing

Given an input and a *way of querying/accessing the input* how many queries to the input are necessary and sufficient to determine (with high probability) whether the input *satisfies a particular property or is "far" from satisfying the property.*

Property Testing

Given an input and a *way of querying/accessing the input* how many queries to the input are necessary and sufficient to determine (with high probability) whether the input *satisfies a particular property or is "far" from satisfying the property.*

Salient features:

- It is easier than answering whether the input satisfies the property or not.
- Usually we are interested in sub-linear query complexity - can't even read the whole input.
- Not too concerned about the time complexity - the main focus is on the query complexity.

Main problems in Field

- Function Property Testing

  Eg: *Test if a given function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *linear.*

  Eg: *Test if a given function* $f : \mathbb{R}^n \to \mathbb{R}$ *monotone.*

  BCGM (Combinatorica 2012), ABCGM (SICOMP 2013), CFM (TOC 2014), ...

- Graph Property Testing

  Eg: *Test if a given graph is* k*-colorable.*

  Eg: *Test if a given graph connected.*

  CFLMN (RANDOM 2007), CFMW (FSTTCS 2010), BC(ToCT 2017), ...

- Geometric Property Testing

  Eg: *Test if a given collection of points in* $\mathbb{R}^n$ k*-clustarable.*

  CPRS (LATIN 2014)

- Distribution Property Testing

  Eg: *Test if a distribution on a domain of size* n *is uniform.*

  Eg: *Test if two distributions are independent.*

  CFMG (SICOMP 2016), CKP (IPL 2016), ...

Main problems in Field

- Function Property Testing

Eg: *Test if a given function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *linear.*

Eg: *Test if a given function* $f : \mathbb{R}^n \to \mathbb{R}$ *monotone.*

BCGM (Combinatorica 2012), ABCGM (SICOMP 2013), CFM (TOC 2014), ...

- Graph Property Testing

  Eg: *Test if a given graph is* k*-colorable.*

  Eg: *Test if a given graph connected.*

  CFLMN (RANDOM 2007), CFMW (FSTTCS 2010), BC(ToCT 2017), ...

- Geometric Property Testing

  Eg: *Test if a given collection of points in* $\mathbb{R}^n$ k*-clustarable.*

  CPRS (LATIN 2014)

- Distribution Property Testing

  Eg: *Test if a distribution on a domain of size* n *is uniform.*

  Eg: *Test if two distributions are independent.*

  CFMG (SICOMP 2016), CKP (IPL 2016), ...

Main problems in Field

- Function Property Testing

  Eg: *Test if a given function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *linear.*

  Eg: *Test if a given function* $f : \mathbb{R}^n \to \mathbb{R}$ *monotone.*

  BCGM (Combinatorica 2012), ABCGM (SICOMP 2013), CFM (TOC 2014), ...

- Graph Property Testing

  Eg: *Test if a given graph is k-colorable.*

  Eg: *Test if a given graph connected.*

  CFLMN (RANDOM 2007), CFMW (FSTTCS 2010), BC(ToCT 2017), ...

- Geometric Property Testing

  Eg: *Test if a given collection of points in* $\mathbb{R}^n$ *k-clustarable.*

  CPRS (LATIN 2014)

- Distribution Property Testing

  Eg: *Test if a distribution on a domain of size n is uniform.*

  Eg: *Test if two distributions are independent.*

  CFMG (SICOMP 2016), CKP (IPL 2016), ...

Main problems in Field

- Function Property Testing

  Eg: *Test if a given function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *linear.*

  Eg: *Test if a given function* $f : \mathbb{R}^n \to \mathbb{R}$ *monotone.*

  BCGM (Combinatorica 2012), ABCGM (SICOMP 2013), CFM (TOC 2014), ...

- Graph Property Testing

  Eg: *Test if a given graph is* k*-colorable.*

  Eg: *Test if a given graph connected.*

  CFLMN (RANDOM 2007), CFMW (FSTTCS 2010), BC(ToCT 2017), ...

- Geometric Property Testing

  Eg: *Test if a given collection of points in* $\mathbb{R}^n$ k*-clustarable.*

  CPRS (LATIN 2014)

- Distribution Property Testing

  Eg: *Test if a distribution on a domain of size* n *is uniform.*

  Eg: *Test if two distributions are independent.*

  CFMG (SICOMP 2016), CKP (IPL 2016), ...

Main problems in Field

- Function Property Testing

  Eg: *Test if a given function* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *linear.*

  Eg: *Test if a given function* $f : \mathbb{R}^n \to \mathbb{R}$ *monotone.*

  BCGM (Combinatorica 2012), ABCGM (SICOMP 2013), CFM (TOC 2014), ...

- Graph Property Testing

  Eg: *Test if a given graph is* k*-colorable.*

  Eg: *Test if a given graph connected.*

  CFLMN (RANDOM 2007), CFMW (FSTTCS 2010), BC(ToCT 2017), ...

- Geometric Property Testing

  Eg: *Test if a given collection of points in* $\mathbb{R}^n$ k*-clustarable.*

  CPRS (LATIN 2014)

- Distribution Property Testing

Eg: *Test if a distribution on a domain of size* n *is uniform.*

Eg: *Test if two distributions are independent.*

CFMG (SICOMP 2016), CKP (IPL 2016), ...

Importance of Property Testing

- Has similarity to statistical sampling - the main difference being the combinatorial structure of the inputs.

- Is closely related to many areas in theory - *communication complexity, Probabilistic Checkable Proofs (PCP), evasiveness theory, learning theory, coding theory (locally decodable codes), approximation theory,* and many more.

- Recently property testing is being used for many real life applications - Google uses it for storing and recovering emails, uses to to understand the network of its users, etc.

One Interesting Problem: Distribution Testing

Given access to a distribution on domain of size n how many queries one need to test if the distribution has a certain property or is "far" from having the property.

One Interesting Problem: Distribution Testing

Given access to a distribution on domain of size n how many queries one need to test if the distribution has a certain property or is "far" from having the property.

Eg:, the simplest and the most fundamental property to test is

"Is a distribution on $\{1, \ldots, n\}$ uniform on the domain?"

One Interesting Problem: Distribution Testing

Given access to a distribution on domain of size n how many queries one need to test if the distribution has a certain property or is "far" from having the property.

Eg:, the simplest and the most fundamental property to test is

"Is a distribution on $\{1, \ldots, n\}$ uniform on the domain?"

- The distribution may be implicitly given.
- Usually a query means drawing a random sample according to the distribution.
- Usually "far" means far in the variation distance.

HOW MANY QUERIES ARE NECESSARY AND SUFFICIENT?

Importance of Distribution Testing

- Distribution Testing is often the central problem in many algorithms and protocols.

Importance of Distribution Testing

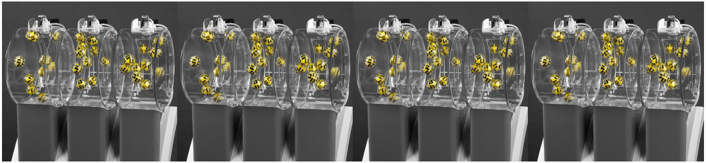
- Distribution Testing is often the central problem in many algorithms and protocols.
- The real life application is huge.

Importance of Distribution Testing

- Distribution Testing is often the central problem in many algorithms and protocols.
- The real life application is huge.
  1. Given a CNF formula producing a satisfying assignments uniformly at random from the set of all satisfying assignments is a crucial problem in the SAT-solver community.
     How to check if the output of a "claimed algorithm" is according to the uniform distribution?

Importance of Distribution Testing

- Distribution Testing is often the central problem in many algorithms and protocols.
- The real life application is huge.
  1. Given a CNF formula producing a satisfying assignments uniformly at random from the set of all satisfying assignments is a crucial problem in the SAT-solver community.
     How to check if the output of a "claimed algorithm" is according to the uniform distribution?
  2. Checking if a random number generator is correct.



Classical Sampling

- The queries are sample drawn according to the distribution
- “far” means total variation distance or the ℓ_2 distance.

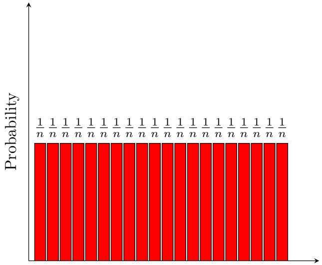


Figure: Uniform Distribution

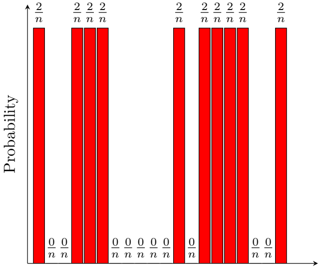


Figure: 1/2-far from uniform

Classical Sampling

- The queries are sample drawn according to the distribution
- "far" means total variation distance or the ℓ_2 distance.

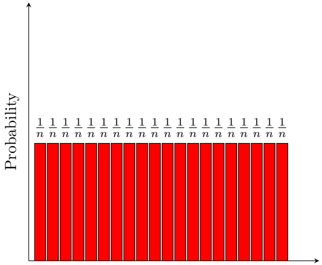


Figure: Uniform Distribution

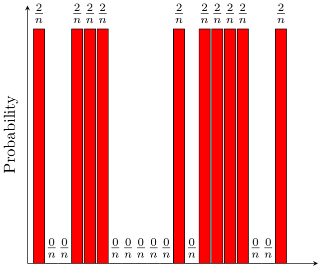


Figure: 1/2-far from uniform

- If $< \sqrt{n}/100$ samples are drawn then with high probability you see only distinct samples from either distribution.

Although sublinear, not practical

Theorem (Batu-Fortnow-Rubinfeld-Smith-White (JACM 2013))

Testing whether a distribution is close to uniform has query complexity $\widetilde{\Theta}(\sqrt{n})$.

Although sublinear, not practical

Theorem (Batu-Fortnow-Rubinfeld-Smith-White (JACM 2013))

Testing whether a distribution is close to uniform has query complexity $\widetilde{\Theta}(\sqrt{n})$.

1. Given a CNF formula producing a satisfying assignments uniformly at random from the set of all satisfying assignments is a crucial problem in the SAT-solver community.
   How to check if the output of a "claimed algorithm" is according to the uniform distribution?

Although sublinear, not practical

Theorem (Batu-Fortnow-Rubinfeld-Smith-White (JACM 2013))

Testing whether a distribution is close to uniform has query complexity $\widetilde{\Theta}(\sqrt{n})$.

1. Given a CNF formula producing a satisfying assignments uniformly at random from the set of all satisfying assignments is a crucial problem in the SAT-solver community.
   How to check if the output of a "claimed algorithm" is according to the uniform distribution?
   Typically, number of variables in the formula ≈ 1000 and the number of satisfying assignments $\approx 2^{70}$.

Although sublinear, not practical

Theorem (Batu-Fortnow-Rubinfeld-Smith-White (JACM 2013))

Testing whether a distribution is close to uniform has query complexity $\widetilde{\Theta}(\sqrt{n})$.

1. Given a CNF formula producing a satisfying assignments uniformly at random from the set of all satisfying assignments is a crucial problem in the SAT-solver community.
   How to check if the output of a "claimed algorithm" is according to the uniform distribution?
   Typically, number of variables in the formula ≈ 1000 and the number of satisfying assignments $\approx 2^{70}$.

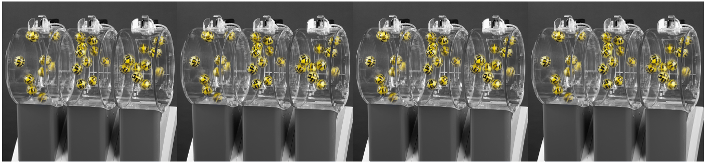
Number of queries needed is around 2^{35}.

Although sublinear, not practical

Theorem (Batu-Fortnow-Rubinfeld-Smith-White (JACM 2013))

Testing whether a distribution is close to uniform has query complexity $\widetilde{\Theta}(\sqrt{n})$.

2. Checking if a random number generator is correct.

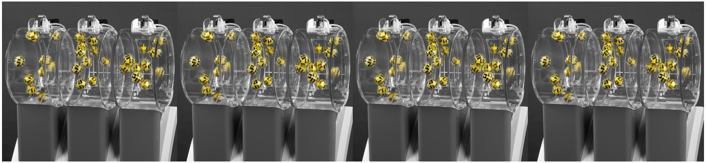


Although sublinear, not practical

Theorem (Batu-Fortnow-Rubinfeld-Smith-White (JACM 2013))

Testing whether a distribution is close to uniform has query complexity $\widetilde{\Theta}(\sqrt{n})$.

2. Checking if a random number generator is correct.



If the machine outputs 12 digit numbers then

Number of times the machine has to be run is 10^6 times.

More sophisticated models of sampling

In the literature more sophisticated models of querying has been studied...

More sophisticated models of sampling

In the literature more sophisticated models of querying has been studied...

- *Quantum queries*

Theorem (C-Fischer-Matsliah-Wolf (2011))

Testing if a distribution on domain size n is uniform has quantum query complexity $\widetilde{\Theta}(n^{1/3})$.

More sophisticated models of sampling

In the literature more sophisticated models of querying has been studied...

- *Quantum queries*

Theorem (C-Fischer-Matsliah-Wolf (2011))

Testing if a distribution on domain size n is uniform has quantum query complexity $\widetilde{\Theta}(n^{1/3})$.

- Many more has been studied ...

More sophisticated models of sampling

In the literature more sophisticated models of querying has been studied...

- *Quantum queries*

Theorem (C-Fischer-Matsliah-Wolf (2011))

Testing if a distribution on domain size n is uniform has quantum query complexity $\widetilde{\Theta}(n^{1/3})$.

- Many more has been studied ...

- *Conditional Sampling*
  Introduced independently by C-Fischer-Matsliah-Goldhirsh (SICOMP 2016) and Cannone-Ron-Servedio (SICOMP 2015).

Conditional Sampling

Definition (Conditional Sampling)

Given a distribution $\mathcal{D}$ on a domain D one can

- *Specify a set $S \subseteq D$,*
- *Draw samples according to the distribution $\mathcal{D}|_S$, that is, $\mathcal{D}$ under the condition that the samples belong to S.*

Conditional Sampling

Definition (Conditional Sampling)

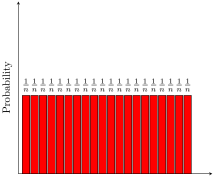
Given a distribution $\mathcal{D}$ on a domain D one can

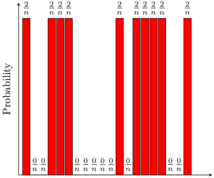
- *Specify a set $S \subseteq D$,*
- *Draw samples according to the distribution $\mathcal{D}|_S$, that is, $\mathcal{D}$ under the condition that the samples belong to S.*

Clearly such a sampling is at least as powerful as drawing normal samples.

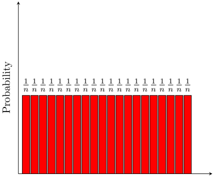
But how much powerful is it?

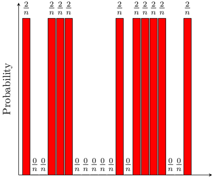
Testing Uniformity Using Conditional Sampling





Testing Uniformity Using Conditional Sampling

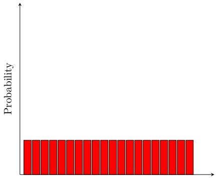


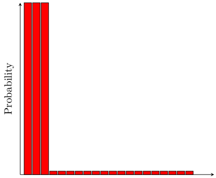


An algorithm for testing uniformity using conditional sampling:

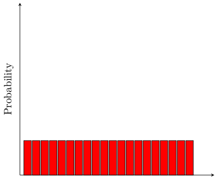
1. Draw two elements x and y uniformly at random from the domain. Let $S = \{x, y\}$.
2. In the case of the "far" distribution, with probability $1/4$, one of the two elements will have probability 0, and the other probability non-zero.
3. Now a constant number of conditional samples drawn from $\mathcal{D}|_S$ is enough to identify that it is not uniform.

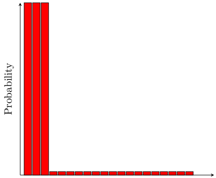
How about other distributions?





How about other distributions?

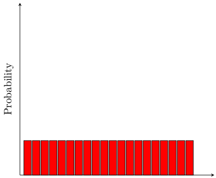


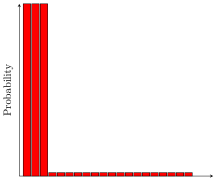


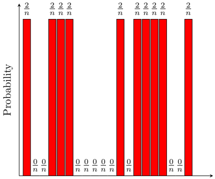
Previous algorithm fails in this case:

1. Draw two elements x and y uniformly at random from the domain. Let $S = \{x, y\}$.
2. In the case of the "far" distribution, with probability almost 1, both the two elements will have probability same, namely ϵ.
3. Probability that we will be able to distinguish the far distribution from the uniform distribution is very low.

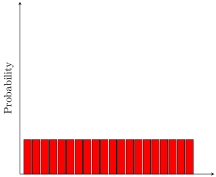
Testing Uniformity Using Conditional Sampling

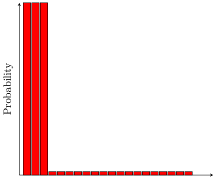


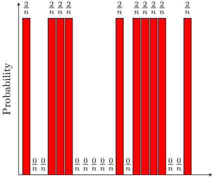




Testing Uniformity Using Conditional Sampling







An algorithm for testing uniformity using conditional sampling:

1. Draw x uniformly at random from the domain and draw y according to the distribution $\mathcal{D}$. Let $S = \{x, y\}$.
2. In the case of the "far" distribution, with constant probability, x will have "low" probability and y will have "high" probibility.
3. We will be able to distinguish the far distribution from the uniform distribution using constant number of conditional samples from $\mathcal{D}|_S$.
4. The constant depend on the farness parameter.

Power of Conditional Samples

Theorem (C-Fischer-Matsliah-Goldhirsh (SICOMP 2016))

For testing if a distribution is uniform one needs only constant number of conditional samples, where constant depends on the distance parameter and the confidence of the algorithm.

Power of Conditional Samples

Theorem (C-Fischer-Matsliah-Goldhirsh (SICOMP 2016))

For testing if a distribution is uniform one needs only constant number of conditional samples, where constant depends on the distance parameter and the confidence of the algorithm.

Theorem (C-Fischer-Matsliah-Goldhirsh (SICOMP 2016))

If $\mathcal{P}$ is any label invariant property then testing whether a distribution satisfy the property $\mathcal{P}$ requires only constant number of condtional samples.

Application to Real Life Problems

The main challenge for using conditional sampling in real life is:
Is conditional sampling implementable?

Application to Real Life Problems

The main challenge for using conditional sampling in real life is:
Is conditional sampling implementable?

Recall the random satisfying assignment problem:

1. Given a CNF formula producing a satisfying assignments uniformly at random from the set of all satisfying assignments is a crucial problem in the SAT-solver community.
   How to check if the output of a "claimed algorithm" is according to the uniform distribution?

Application to Real Life Problems

The main challenge for using conditional sampling in real life is:
Is conditional sampling implementable?

Recall the random satisfying assignment problem:

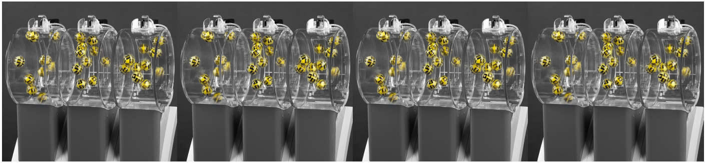
1. Given a CNF formula producing a satisfying assignments uniformly at random from the set of all satisfying assignments is a crucial problem in the SAT-solver community.
   How to check if the output of a "claimed algorithm" is according to the uniform distribution?

Theorem (C-Meel-Vardi (ongoing project))

Using conditional sampling we can design a "practical" algorithm that given black box access to an algorithm can test if the algorithm indeed performs the task properly.

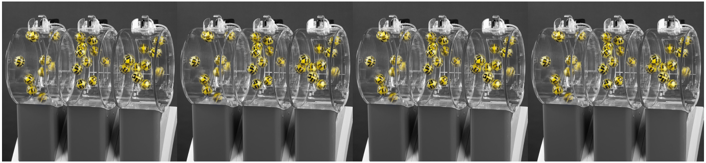
How about the Lottery Machine Problem

2. Checking if a random number generator is correct.



How about the Lottery Machine Problem

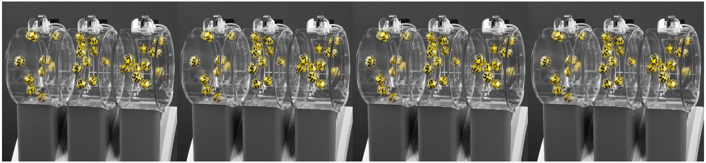
2. Checking if a random number generator is correct.



- In Lottery Machine we can easily draw conditional samples when the set S is of the form $S_1 \times S_2 \times \ldots$, where the $S_i \subseteq \{0, 1, \ldots, 9\}$.

How about the Lottery Machine Problem

2. Checking if a random number generator is correct.



- In Lottery Machine we can easily draw conditional samples when the set S is of the form $S_1 \times S_2 \times \ldots$, where the $S_i \subseteq \{0, 1, \ldots, 9\}$.
- But recall that for our algorithm for testing uniformity the set S can be any two elements in the domain and not necessarily of the special structure for which we can execute the conditional samples.

Conditional Sampling on Structured Domain

Theorem (Bhattacharyya-C (ArXiv 2017, submitted))

When the domain is of the form $D_1 \times D_2 \times \cdots \times D_m$ and conditional sample can be drawn for sets of form $S_1 \times \cdots \times S_n$, where $S_i \subseteq D_i$, then $\Omega(m)$ number of conditional samples are necessary and $O(m^2)$ number of samples are sufficient.

- This has applications in Cryptography and Derandomization.

Further Works

Many different extension and directions are being investigated. For example:

- *Quantum Conditional Sampling* (Sardharwalla-Strelchuk-Jozsa (QIC 2016)).
- *Big data* (Canonne-Rubinfeld (ICALP 2014))
- *Learning using Conditional Sampling* (Aliakbarpour-Blais-Rubinfeld (COLT 2016))

A good survey for this area is "A Chasm Between Identity and Equivalence Testing with Conditional Queries" by Jayadev Acharya, Clément L. Canonne and Gautam Kamath.

Future Direction

Future Direction

- One of the most important problem of the present and future is testing if an algorithm is correct. Property testing has a big role to play.

Future Direction

- One of the most important problem of the present and future is testing if an algorithm is correct. Property testing has a big role to play.
- Major challenge: To model the problems properly.

Future Direction

- One of the most important problem of the present and future is testing if an algorithm is correct. Property testing has a big role to play.
- Major challenge: To model the problems properly.
- Interesting progress has been made in recent times on applying property testing successfully to real life problems.

Future Direction

- One of the most important problem of the present and future is testing if an algorithm is correct. Property testing has a big role to play.
- Major challenge: To model the problems properly.
- Interesting progress has been made in recent times on applying property testing successfully to real life problems.
- Lots of interesting and challenging theoretical problems arise.

About Me

Job Experience

- Phd Students at University of Chicago (2003-2008)
- Postdoc at Technion Instidute of Technology (2008-2009)
- Postdoc at Centrum Wiskunde Informatica (CWI, Amsterdam) (2009-2010)
- Faculty at Chennai Mathematical Institute (CMI) (2010-present)
- Visiting Faculty at University of California, San Diego (UCSD) (2014 - for 6 months)
- Visiting Faculty at Centrum Wiskunde Informatica (CWI, Amsterdam) (2017 - for 1 year)
- Multiple short term (1 to 3 months) visits to Technion Institute of Technology, Simon Institute, Max Plank Institute (MPI, Saarbruken), University of Chicago, Rice University, MSR Bangalore, University of Paris 7.
- Many short term visits to various universities.

My areas of research

My areas of research

- Property Testing: Query/Sample Complexity for promise problems.

My areas of research

- Property Testing: Query/Sample Complexity for promise problems.
- Fourier analysis of Boolean functions: Understand the Fourier Spectrum of a Boolean function.

My areas of research

- Property Testing: Query/Sample Complexity for promise problems.
- Fourier analysis of Boolean functions: Understand the Fourier Spectrum of a Boolean function.
  For example: Trying to answer the Fourier Entropy Influence Conjecture.

My areas of research

- Property Testing: Query/Sample Complexity for promise problems.
- Fourier analysis of Boolean functions: Understand the Fourier Spectrum of a Boolean function.
  For example: Trying to answer the Fourier Entropy Influence Conjecture.
- Quantum Computing

My areas of research

- Property Testing: Query/Sample Complexity for promise problems.
- Fourier analysis of Boolean functions: Understand the Fourier Spectrum of a Boolean function.
  For example: Trying to answer the Fourier Entropy Influence Conjecture.
- Quantum Computing
- Online/streaming algorithms:
  For example: The matroid secretary problem.

My areas of research

- Property Testing: Query/Sample Complexity for promise problems.
- Fourier analysis of Boolean functions: Understand the Fourier Spectrum of a Boolean function.
  For example: Trying to answer the Fourier Entropy Influence Conjecture.
- Quantum Computing
- Online/streaming algorithms:
  For example: The matroid secretary problem.
- Graph algorithms.

My areas of research

- Property Testing: Query/Sample Complexity for promise problems.
- Fourier analysis of Boolean functions: Understand the Fourier Spectrum of a Boolean function.
  For example: Trying to answer the Fourier Entropy Influence Conjecture.
- Quantum Computing
- Online/streaming algorithms:
  For example: The matroid secretary problem.
- Graph algorithms.
- Algorithmic game theory.

My areas of research

- Property Testing: Query/Sample Complexity for promise problems.
- Fourier analysis of Boolean functions: Understand the Fourier Spectrum of a Boolean function.
  For example: Trying to answer the Fourier Entropy Influence Conjecture.
- Quantum Computing
- Online/streaming algorithms:
  For example: The matroid secretary problem.
- Graph algorithms.
- Algorithmic game theory.
- Information theory, Coding theory and cryptography

My areas of research

- Property Testing: Query/Sample Complexity for promise problems.
- Fourier analysis of Boolean functions: Understand the Fourier Spectrum of a Boolean function.
  For example: Trying to answer the Fourier Entropy Influence Conjecture.
- Quantum Computing
- Online/streaming algorithms:
  For example: The matroid secretary problem.
- Graph algorithms.
- Algorithmic game theory.
- Information theory, Coding theory and cryptography
- Combinatorial problems

Experiences in Research Guidance

- 2 students have received their Phd under my guidance:
  1. Rameshwar Pratap (co-advised with Prof. Samir Datta)
     Thesis: Some Problems in Sublinear Algorithms. (2014)
  2. Nitesh Jha Thesis: Finding Transitive Subgraphs and Counting Popular Matchings. (2017)

Experiences in Research Guidance

- 2 students have received their Phd under my guidance:
  1. Rameshwar Pratap (co-advised with Prof. Samir Datta) Thesis: Some Problems in Sublinear Algorithms. (2014)
  2. Nitesh Jha Thesis: Finding Transitive Subgraphs and Counting Popular Matchings. (2017)
- Adviced a number of masters students: Sandipan Bhattacharyya, Biswaroop Maiti, Pranabendu Misra, Shion Samadder Chaudhury, Rupam Acharyya and V. Vivek.

Experiences in Research Guidance

- 2 students have received their Phd under my guidance:
  1. Rameshwar Pratap (co-advised with Prof. Samir Datta) Thesis: Some Problems in Sublinear Algorithms. (2014)
  2. Nitesh Jha Thesis: Finding Transitive Subgraphs and Counting Popular Matchings. (2017)
- Adviced a number of masters students: Sandipan Bhattacharyya, Biswaroop Maiti, Pranabendu Misra, Shion Samadder Chaudhury, Rupam Acharyya and V. Vivek.
- Was involved in projects with few phd students during their phd: David G.Soriano (CWI, Amsterdam), Nitin Saurabh (IMSc), Rishiraj Bhatyacharyya (ISI Kolkata), Srinivasan Arunachalam (CWI, Amsterdam).

Experiences in Research Guidance

- 2 students have received their Phd under my guidance:
  1. Rameshwar Pratap (co-advised with Prof. Samir Datta) Thesis: Some Problems in Sublinear Algorithms. (2014)
  2. Nitesh Jha Thesis: Finding Transitive Subgraphs and Counting Popular Matchings. (2017)
- Adviced a number of masters students: Sandipan Bhattacharyya, Biswaroop Maiti, Pranabendu Misra, Shion Samadder Chaudhury, Rupam Acharyya and V. Vivek.
- Was involved in projects with few phd students during their phd: David G.Soriano (CWI, Amsterdam), Nitin Saurabh (IMSc), Rishiraj Bhatyacharyya (ISI Kolkata), Srinivasan Arunachalam (CWI, Amsterdam).
- Have been in the thesis committee for several students (Phd and masters).

Experiences in Research Guidance

- 2 students have received their Phd under my guidance:
  1. Rameshwar Pratap (co-advised with Prof. Samir Datta) Thesis: Some Problems in Sublinear Algorithms. (2014)
  2. Nitesh Jha Thesis: Finding Transitive Subgraphs and Counting Popular Matchings. (2017)
- Adviced a number of masters students: Sandipan Bhattacharyya, Biswaroop Maiti, Pranabendu Misra, Shion Samadder Chaudhury, Rupam Acharyya and V. Vivek.
- Was involved in projects with few phd students during their phd: David G.Soriano (CWI, Amsterdam), Nitin Saurabh (IMSc), Rishiraj Bhatyacharyya (ISI Kolkata), Srinivasan Arunachalam (CWI, Amsterdam).
- Have been in the thesis committee for several students (Phd and masters).
- Was the faculty adviser for the Masters and Phd students for 2011 to 2014 and 2014 to 2016 respectively at CMI.

Teaching Experiences

- Taught a number of courses at CMI (both undergraduate level and masters level): Discrete Mathematics, Algorithms, TOC, Advanced Algorithms, Coding Theory, Information Theory, Algorithmic Game Theory, Probabilistic Methods, Property Testing, Advanced Graph Algorithms.

Teaching Experiences

- Taught a number of courses at CMI (both undergraduate level and masters level): Discrete Mathematics, Algorithms, TOC, Advanced Algorithms, Coding Theory, Information Theory, Algorithmic Game Theory, Probabilistic Methods, Property Testing, Advanced Graph Algorithms.
- Taught Discrete Mathematics and Combinatorics (Undergraduate level: Basic and Advanced) courses at University of California, San Diego (UCSD). (Over 400 students in each class).

Teaching Experiences

- Taught a number of courses at CMI (both undergraduate level and masters level): Discrete Mathematics, Algorithms, TOC, Advanced Algorithms, Coding Theory, Information Theory, Algorithmic Game Theory, Probabilistic Methods, Property Testing, Advanced Graph Algorithms.
- Taught Discrete Mathematics and Combinatorics (Undergraduate level: Basic and Advanced) courses at University of California, San Diego (UCSD). (Over 400 students in each class).
- Taught a whole summer school (Research Education for Undergraduates, REU) at Univetsity of Chicago in 2014,

Teaching Experiences

- Taught a number of courses at CMI (both undergraduate level and masters level): Discrete Mathematics, Algorithms, TOC, Advanced Algorithms, Coding Theory, Information Theory, Algorithmic Game Theory, Probabilistic Methods, Property Testing, Advanced Graph Algorithms.
- Taught Discrete Mathematics and Combinatorics (Undergraduate level: Basic and Advanced) courses at University of California, San Diego (UCSD). (Over 400 students in each class).
- Taught a whole summer school (Research Education for Undergraduates, REU) at Univetsity of Chicago in 2014,
- Offered many reading courses at CMI, Technion and CWI.

Teaching Experiences

- Taught a number of courses at CMI (both undergraduate level and masters level): Discrete Mathematics, Algorithms, TOC, Advanced Algorithms, Coding Theory, Information Theory, Algorithmic Game Theory, Probabilistic Methods, Property Testing, Advanced Graph Algorithms.
- Taught Discrete Mathematics and Combinatorics (Undergraduate level: Basic and Advanced) courses at University of California, San Diego (UCSD). (Over 400 students in each class).
- Taught a whole summer school (Research Education for Undergraduates, REU) at Univetsity of Chicago in 2014,
- Offered many reading courses at CMI, Technion and CWI.
- Offerd a course on Discrete Mathematics on NPTEL.

Teaching Experiences

- Taught a number of courses at CMI (both undergraduate level and masters level): Discrete Mathematics, Algorithms, TOC, Advanced Algorithms, Coding Theory, Information Theory, Algorithmic Game Theory, Probabilistic Methods, Property Testing, Advanced Graph Algorithms.
- Taught Discrete Mathematics and Combinatorics (Undergraduate level: Basic and Advanced) courses at University of California, San Diego (UCSD). (Over 400 students in each class).
- Taught a whole summer school (Research Education for Undergraduates, REU) at Univetsity of Chicago in 2014,
- Offered many reading courses at CMI, Technion and CWI.
- Offerd a course on Discrete Mathematics on NPTEL.
- Offered a course on Introduction to Algorithms on E-lectures for the UGC for TV viewers.

Other Professional Activities

- Organized a number of Workshops/Conferences

Other Professional Activities

- Organized a number of Workshops/Conferences
- Referreed over 100 papers.

Other Professional Activities

- Organized a number of Workshops/Conferences
- Referreed over 100 papers.
- Have served in a number of committees (both academic and adminstrative) at CMI.

Other Professional Activities

- Organized a number of Workshops/Conferences
- Referreed over 100 papers.
- Have served in a number of committees (both academic and adminstrative) at CMI.
- Wrote a number of scientific/survey articles in magazines intended for general public.

My family