ITCS 2020

D Ellis Hershkowitz

January 21, 2020

These are the notes I took while at the Innovations in Theoretical Computer Science (ITCS) 2020. These notes were written while trying to keep up with the talks and so are not free from errors. Cheers!

Contents

1 January 12, 2020

1.1 Josh Alman on OV graphs are (probably) hard instances

OV Graphs

Defined / inspired by orthogonal vector problem: given $v_1, \ldots, v_n$ d-dimensional vectors; goal is to determine if $\exists i, j$ s.t. $< v_i, v_j >= 0$

Can construct a graph where two nodes are adjacent if their corresponding vectors are orthogonal

Can then ask graph problems re OV graphs:

- OV: Does G have any edges

etc

SAT and OV

OV studied because of nice result; faster algs for OV lead to breakthroughs for SAT

OV conjecture: for every $\epsilon > 0$ there is a $c > 0$ s.t. solving OV requires $\Omega(n^{2-\epsilon})$ time

Stated SETH

Williams showed SETH implies OV conjecture

This Paper

An analogy of this result for many graph problems on OV graphs; show solving their problems on OV graphs then would get faster algorithms for MAX-SAT

Max-k-SAT

Given a k-CNF formula; try to sat max number of clauses

Best algs:

- MAX-2-SAT take $O(2^{\omega}n/3) \leq O(2^{.791n})$
- No non-trivial algorithms for $k \geq 3$

These are even the best algorithms for sparse instances with $O(n)$ clauses

OVΔ

Best alg to find triangles in general graphs takes $O(n^{\omega})$

New theorem says: if can find triangles in $O(n^{\omega-\delta})$ time in OV graphs then can solve $O(n)$ clause MAX-2-SAT in time $O(2^{(\omega/3-\delta)n})$

Similar results for several other OV graph problems

Are all problems hard on OV graphs

Any problem that is NP-hard on sparse graphs is also NP-hard on OV graphs of dimension $d = O(\log n)$

But in general, no: they show max clique solvable in $2^d \cdot \text{poly} n$ time as well as online matrix-vector multiplication

1.2 Joseph Landsberg on Tensors not subject to barriers for Strassen's laser method

Tensor wrt C^9 could be used in "Strassen's laser method" to prove exponent of matrix multiplication is 2

Matrix mult is a bilinear map

Matrix mult viewed as point in tensor space of trilinear maps

Like with matrix rank, rank of tensor is smallest r such that T can expressed as sum of r rank one tensors; also defined **border rank**

Strassen/Bini showed look at border rank function as function of n; growth of this function determines matrix mult exponent

Defined Kronecker powers

Matrix multiplication tensor is invariant(?) under Kronecker powers

Laser method: can degenerate high Kroneckers powers of tensors to a matrix mult tensor to get an upper bound on ω

Most famous tensor is the Coppersmith-Winograd tensor; a new tensor gives $\omega < 2.373$

2014: game is over for the Coppersmith-Winograd tensor

So now want to find other tensors to find better upper bounds

The Little CW Tensor

Simple class of tensors shows $\omega \leq 2.41$

Theorem of this work: bad news for these classes of tensors

Good news: spent some time looking for some tensors; found a new class that is e.g. a skew symmetric version of the above tensors; bad news is it's even worse but get a promising "drop" when you square it

Motivation: wanted to study these tensors not as cominatorial but as geometric objects; wanted to look for tensors with similar geometric properties to the above tensors

When you square these skew symmetric tensors you get back det_3 and per_3

1.3 Artsiom Hovarau On a Theorem of Lovasz that hom(., H) Determines the Isomorphism Type of H

Graph Homomorphisms

Defined $hom(G, H)$ function; in '67 paper Lovasz showed if homomorphisms identical then graphs are isomorphic

Many graphs problems expressible as graph homomorphisms; e.g. vertex-cover is H as an edge with one self-loop

Dichotomies for GH

Known complexity dichotomies for graph homomorphisms

Relevant works

Gave various results on graph homomorphisms

Labeled Graphs

labeled graph has labels on vertices; take union and identify vertices with same labels

Can express graph homomorphisms as linear combination of functions that they(?) define

Lovasz Result

Gave Lovaz and Schrijver results

Their results: a partial graph homomorphism function uniquely preserves the RHS graph

Introduce what they call "Vandermonde" argument

Applications to Complexity given

1.4 Sivara Ramamoorty on Equivalence of Systematic Linear Data Structures and Matrix Rigidity

Cicuits and data structures are important models of comp; this talk about connecting the two

Motivation: recent works show DS lower bounds imply circuit lower bounds

Show "rectangular rigidity" equivalent to "systematic linear data structures"

Rectangular Rigidity

Want a subset of vertices in F_2^n that is "far-away" from any low dimensional subspace

- "small" random sets have this property; where small means you can't express vectors as sum of low rank and row sparse matrix

This is a generalization of a notion of Valiant

Formally, Q is (r, t)-rigid if every r-dim subspace has a point at hamming distance at least t from V

Question: relationships between r, t, $|Q|$ and n; gave known lower bounds for explicit Q

Systematic Linear Model

Store an $x \in F_2^n$; want to compute $< q, x >$ queries

An Upper Bound

An upper bound if Q is not rigid; express Q as sum of rank r A and row sparse B; then can compute these queries

Also show other direction; if set is rigid then a lower bound on query time

Explicit Set Q

For this version show similar results

1.5 Adam Polak on Monochromatic triangles, intermediate matrix products, and convolutions

Talk is about matrix multiplication; gave series of improvements for ω up to current $O(n^{2.3728639})$

What if we want to compute a matrix product over something other than a ring; e.g. $(\min, +)$-product which is runtime equivalent to all pairs shortest paths (in a weighted graph)

Conjectured that exponent for this must be at least 3

So there is an "easy" and "hard" matrix multiplication

There are problems in the middle which require $O(n^{(\omega+3)/2})$ time (call these “intermediate” matrix product problems)

Wide open question: is the same running time of all these problems a coincidence?

This work gives some reductions in this area

Convolutions

Can do $O(n \log n)$ for easy via Fourier

$\tilde{O}(n^{3/2})$ for intermediate

$O(n^{2-o(1)})$ for hard

Give reductions in this paper for these sorts of problems; gave some highlights of reductions

Sketch

Min-Max in $T(n)$ time gives unweighted APSP in $T(n) \log n$ time

Process input matrix in $\log n$ rounds; have matrix of shortest paths of length up to 2^i

Process even and odd paths differently via middle-first-search a la Savitch’s theorem sort of thing

1.6 Nima Anari on Matching is as Easy as the Decision Problem, in the NC Model

PM Problem

Find a PM in NC model

Two natural questions

1. $\exists$ PM
2. Output a PM

In fact, study weighted analogues of these

Show there is a fast, parallel deterministic reduction from search-to-decision problem (provided weights are poly bounded)

Motivation for NC

Two classic algorithms for PM:

1. Augmenting paths / polyhedral (deterministic but sequential)

2. Determinant of matrices / algebraic (randomized but parallel since det is fast in PRAM)

Question: can we get the best of both worlds? I.e. a deterministic, parallel algorithm

No such algorithm is known in NC but is known in randomized NC

Also quasi-NC where allowed $n^{\text{poly}\log n}$ processors; recent work shows decision and search in general graphs for this model

Today: pseudo-deterministic NC; can use randomization but output should be unique function of input w.h.p.; past work shows possible in bipartite; this work shows also true of general graphs

Another corollary: if graph is minor-closed then counting implies search

Bits of Algorithm

Use notion of **matching minor**

Edmonds showed matching polytope is integral; says must escape all odd sets

If looking at a face of polytope; this is defined by a set of disallowed edges and a tight odd set; the tight odd set can be contracted

So can choose a weight function and reduce input problem by then removing disallowed edges and contracting tight odd sets; have to carefully choose weight functions

Key lemma is showing there exists a way to choose weight function that removes a $1/\text{poly}\log n$ fraction of edges

Future work: get reduction to work for unweighted graphs

1.7 Gal Yona on Preference-Informed Fairness

Collection of individuals X and collection of outcomes C (e.g. yes/no for receiving a loan)

Objective: map individuals to outcomes to provide protections against discrimination

This Work

New notion of fairness: **preference-informed individual fairness**; a relaxation of individual fairness and envy-freeness

Then study problems subject to PIIF

Individual fairness

Want to treat "similar individuals similarly"

Map individuals to outcomes space; individually fair if every two indivuduals distance between their outcome distributions is similar to their distance

E.g. two equally-credit worthy individuals should get loan according to similar probabilities

Allocation which gives individuals their favorite outcome is not individually fair; this notion of similar treated similarly is over-resetrictive in this sense

Envy Freeness

From fair division literature

Every individual prefers their outcome to those of all other individuals

E.g. allocation which gives people their favorite outcome is envy-free

At same time if two individuals want the same thing but don't both get it then it's not envy free even if they are not similarly qualified

So envy-freeness is overly restrictive in this sense

This work interpolates between both; ask counterfactural of under IF you would receive blah, do you prefer your current outcome to this?

Show that this generalizes both IF and EV and also that IF $\neq$ EV

1.8 Saeed Sharifi-Malvajerdi on A New Analysis of Differential Privacy's Generalization Guarantees

Reproducibility Crisis (Overfitting)

To avoid overfitting classical statistics suggests you fix your queries before looking at the data set; but in practice usually first look at the data and adapt your queries to the data and iterate

Adaptive Data Analysis

Question is how to perform valid statistical analysis in the adaptive model

A naive solution: sample splitting

- Partition data into k parts where each answers one of your k adaptive queries; this is suboptimal

Recent line of works show how to improve on this approach using **differential privacy**

Major theorem: transfer theorem; "show in-sample accuracy + DP imply out-of-sample accuracy"

These achieve the optimal rate of $n = \Omega(\sqrt{k})$

This Talk

A new proof of the transfer theorem which is

- Simpler
- Gives new structural insights
- Concrete bounds are better re constants

Differential Privacy

A property of a randomized algorithm where output distribution does not change if you change one data point in your data set

Usually attained by adding noise to computations; e.g. to compute an average just add a little bit of noise

Can also see DP as a "stability" mechanism; i.e. not sensitive to individual data points

Model of Adaptive Analysis

A randomized mechanism queries from a data set and gets to adaptively decide its queries

Want answers given by algorithm are close to the value of the queries on the distribution

Stated Transfer Theorem

Proof Sketch

In-sample accuracy implies (on its own) answers of the algorithm are close to the value of the query when sample from conditional distribution of data se

DP on its own implues that value of query on conditional distribution is close to value of distribution on P

Rest is just triangle inequality

1.9 Martin Hoefer on Strategic Payments in Financial Networks

Financial networks and Systemic risk

Findancial entities and liabilities and dependencies

Crisis of '08

Ongoing challenge: debts as a major source of risk

Main goal: understand effects and design regulation to avoid cascading insolvency

Network Model

Main model of modeling risk; financial institutions are vertices; relations are arcs; edges have value of debt between vertices; each institution has external assets

Question is in this system who is bankrupt; i.e. who can payoff debt

Idea is to design a money flow

Total assets of an institution are how much their owed plus their internal assets

Assume debts are payed proportional to what the institution owes (pro rata)

Question of this paper: what about **incentives**?

Strategic Payments

What are strategic incentives in how to pay debts and how does this affect insolvency? I.e. replace pro rata with incentives

Some natural assumptions about how debts are paid; e.g. if have money must pay debts

Seniorities: a priority order in which debts must be cleared; can do **edge** or **coin** or **general** strategies

Strategy of firms is to maximize their total assets; so given a strategy profile want to determine the assets

This paper focuses on unique component-wise maximal clearing state for edge-ranking games

Edge Ranking Results

Pure national equilibria may not exists

Computing most natrual things is NP-hard

Computing total social welfare has unbounded price of anarchy

Coin-Ranking Results

Things are much better

E.g. compute strong equilibrium in poly time etc.

Some open problems given

1.10 Siddharth Prasad on Incentive Compatible Active Learning

Motivation

A model of active learning that takes into account incentive issues

In economic experiments: learner wants to elicit parameters governing their preferences

Experiments always incentivized

In active learning learner gets to choose data points where they want to see a label

Preference elicitation Setup

Agents have types drawn from metric type space

Some abstract outcome spsace

Learner executes learning algorithm to learn agent's true type

An agent can be **strategic** in this interaction; maybe an agent wants to play according to a strategy to steer interaction towards higher utility

Can combine learning and incentive compatibility to get "IC" complexity

A Simple Method

Could assign to every type an outcome where how much you like a type is proportional to distance from your true type

Gave a concrete example which can be very inefficient

A sufficient condition for when you can do this

Theorem: If all upper contour sets satisfy some properties can run the simple metho

1.11 Lior Goldberg on DEEP-FRI: Sampling Outside the Box Improves Soundness

A low degree testing protocol

Question

Have Reed-Solomon code

FRI

FRI: fast RS interactive oracle proof of proximity

Want to distinguish between f in the code and f at least δ-far from the code

Work in IOPP; in each round prover sends an oracle to the verifier

Goal: reduce size of problem by 2:

1. split polynomial P into two polynomials of half the degree
2. Take D a miltiplicative group of size 2^k
3. Smaller domain, same rate
4. Merge two polynomials using random linear combinations

1.12 Hard properties with (very) short PCPPs and their applications

PCPs

Verifier reads statement x, tosses coins to determine query locations, reads $O(1)$ locations and accepts/rejects

1. If x true the there's some proof so that verifier always accepts
2. If x false then verifier rejects w.h.p.

PCPPs

One way to construct short PCPs is by using PCPPs

This time verifier doesn't read in all of x but indexes into it at randomly chosen query locations

1. if x then there is some proof that causes accept
2. If x is very far from attaining the property verifier rejects w.h.p.

Can find properties with PCPPs with only a poly $\log$ overhead; question is can you find languages with PCPPs with only a constant factor overhead; recently show that can

Question is can find language L without constant query and constant overhead; a good candidate is a property that is hard to test in the property testing setting ; i.e. is poly $\log$ overhead necessary or a property that requires looking at a constant fraction of input to test

Results

For every l there's a property s.t.

1. Any property tester requires $\Omega(n)$ queries
2. Property has constant query PCPP with proof length $O(n \log^l n)$

Use this to show relationship between PT and "tolerant" PT; tolerant PT is like PT but where have separate ϵ and ϵ' where if within ϵ' must accept (more difficult than PT)

Prior Work

There is a property s.t. testing it requires constant queries but a tolerant tester requires near-linear queries; this work's result improves the "near" in "near-linear"

Proof Idea

Append PCPP proof of property for y to string y

1.13 Elizabeth Yang on High-Dimensional Expanders from Expanders

Motivation

HDXes are richly structured objects with connections to other areas of TCS; gave examples

Don't know too much about constructions of HDXs

Currently no known combinatorial or randomized constructions

High-Dimensional Graphs (Simplixial Complexes)

In addition to vertices and graphs have k-faces, each containing $k+1$ vertices

Have to have downward closure (i.e. all subsets of a face)

Every face assigned a weight and its weight is the sum of the weights of its parents

k-Down-Up Random Walk

States: k-faces

Transitions via $(k-1)$-faces

Transition down uniformly then up proportional to weights

An HDX means k-DU walks have spectral gap that only depends on k

Contribution

Construct an inf family of constant degree HDXs whose k-DU walks have a spectral gap only depending on k

Inherit some nice properties from expanders

Construction

Take T-regular expander; add self-loops; take tensor product of this graph with complete graph

The faces are the cliques of this graph

After taking tensor product, put uniform weights on $(H+1)$-cliques

Cliques can span across an edge of G

3 Ways to Transition

1. Move from one edge of G to another
2. Change numbers belong in your k-face
3. Change u, v labels within an edge without changing numbers

Projection and Restriction Chains

Start with a Markov chain; partition states into "restriction chains"; which also induces a projection chain which tells you how to move within restriction chains

Spectrail gap of chain is lower bounded by product of projection and restriction spectral gaps

Two Types of Faces

Pure: all labels in one G vertex

Mixed: not

Not an obvious symmetric way of deciding what the restriction chains; shows how where edges of G correspond to restriction chain

Then have to apply rest/proj chain idea again

1.14 Peter Manohar On Local Testability in the Non-Signaling Setting

Non-Signaling Functions

aka Sherali-Adams

A k-non-signaling $\mathcal{F}$ is like a quantum function in the sense that the evaluation procedure is probabilistic: choose inputs S; feed them in and get back evaluation on these; but can only look on $\mathcal{F}$ on at most k points

Motivation

nsPCPs have applications to classical computer science: crypto and complexity

Results

Study non-signaling locally testable codes, focusing on low-degree tests

Show that evenly-spaced degree-d test against non-signaling functions works for large k and not for smaller k

Also show a **result 2**

Defined linear codes and dual code

Goal: establish a non-signaling analogue of classic linear/dual code fact

Have to define what it means for $\mathcal{F}$ to be in C also what inner product wrt dual code means; do so

Summary

First result

Second result shows that T is an l-local characterization of C iff T proves $C^{\perp}$

2 January 13, 2020

2.1 Wei-Kai Lin on MPC for MPC: Secure Computation on a Massively Parallel Computing Architecture

Models of Parallel Computation

Want to make secure parallel computation

Many models, consider the MPC model here

MPC

m Random Access Machines (RAM); network is fully connected; each of N machines has space s

Here $s = N^{\epsilon}$

MPC Proceeds in Rounds

Nodes alternate rounds local computation and sending messages (still with space constraint)

At the end all machines jointly output their answer; the metric is the number of rounds

Compared MPC to PRAM; e.g. MPC can use $O(1)$ rounds to sort but PRAM needs $\Omega(\log n)$

Main question: how to get MPC algorithms "secure"? What is the cost?

Many adversarial settings you could use to define "secure"

In crypto MPC usually stands for secure Multi-Party Computation

Two Adversarial Scenarios

In first adversary sees every message on the network; wants to learn secret input

In this scenario show the round blowup is constant

In second model adversary corrupts some machines, wants to learn secret of others

In this scenario show round blowup is poly$k$ where k is security parameter

E.g. of First Scenario

Technique: oblivious routing

Want an oblivious routing algorithm that's capable of sending messages to their prescribed receiver without revealing to the adversary the prescribed receiver

Use the well-known butterfly network; has nice property for any input/output pair there is an easy-to-compute path

Routing from butterfly network then uses two steps; in first send to a random intermediate node using the butterfly network; then use another butterfly network to send from the intermediate node to prescribed receiver; get security via simple Chernoff bound

Only problem is number of layers in butterfly network is logarithmic in num machines which means logarithmic blowup in rounds

Idea to fix problem: merge several layers of the butterfly network

2.2 Nathan Harms on Universal Communication, Universal Graphs, and Graph Labeling

The Point of This Talk

Field of communication complexity on left and on right field of graph labeling and universal graphs; these fields are disjoint; today wants to talk about stuff in the intersection

Universal Graphs

A family of graphs $\mathcal{F}$; a graph U is universal for $\mathcal{F}$ if for all graphs in G there is a mapping from vertices in G to U that maintains vertex adjacency; that is, U has as a subgraph every graph of $\mathcal{F}$

Graph Labeling

Again family of graphs $\mathcal{F}$

A function h that takes in a graph and two of its vertices

Goal: find a decoder D s.t. for all $G \in \mathcal{F}$, should be able to assign a short label $l(v)$ such that for every pair of vertices x, y we can get function value from labeling

E.g. small distance labelings, where fix k and function is indicator of whether vertices are within distance k; labeling can be the adjacency labeling

Here labeling implies universal graphs

Communication Model

What model of communication looks the most like this model? Simultaneous message passing:

- Have Alice and Bob and a ref; A and B send a single message to ref based on their private inputs that has to output $h(x, y)$

Natural questions

1. What if ref doesn't know h; e.g. A and B receive h as input?
2. What if A and B are in some shared environment that ref doesn't know

E.g. A and B in same city; ref needs to decide if they are nearby without knowing their locations

To answer these questions can expand this to the universal SMP model: A and B receive graph from $\mathcal{F}$ and inputs x and y; the ref has to compute $h(x, y, G)$

An Example

Let $\mathcal{F}$ be graphs of degree at most 5; let $h(x, y, G)$ be adjacency in G

1. Alice sends each neighbor of x; Bob sends y
2. Ref checks if y is neighbor of x

Cost: $O(\log n)$; randomized cost is $O(1)$

SMP, USMP and Graph Labeling

Easy to see USMP generalizes SMP

Also USMP generalizes graph labeling; challenge is A and B might have different functions so need to make this protocol symmetric

Thus, USMP generalizes SMP and graph labeling; thus something in the middle as stated at the beginning

Facts About USMP

Classic theorem of communication complexity lets you derandomize (Newman's theorem) with a log blowup (though this is non-constructive)

Also a 2-way protocol can be made into a universal protocol

Together this means 2-way randomized protocols give upper bounds on graph labeling problems

Some Other Results

k-Hamming Distance Protocols give poly size universal graph for n-vertex induced subgraphs of hypercubbe

Also $O(\log n)$ cost labeling schemes for $dist_k$ on distributive lattices

Also $O(k)$ cost USMP protocol for $dist_k$ on trees

Also $O(1)$-cost USMP protocol for $dist_2$ on planar graphs

Also interval graphs equivalent to greater-than

Gave some open problems

2.3 Yael Tauman Kalai on Interactive Coding with Constant Round and Communication Blowup

About interactive coding; a generalization of standard ECCs

Main error models considered with ECCs are stochastic and adversarial; here they consider adversarial; they consider ins+del adversaries

A good ECC is resilient to constant fraction of adversarial error and has efficient encoding and decoding

This Talk

Consider interactive coding where a conversation back and forth; want to compile this to a new conversation that is resilient to errors

Here good coding scheme is similar to good in ECC case

Trivial solution is to just apply an ECC message-by-message; problem is this is not resilient to constant fraction of adversarial error; e.g. adversary can totally corrupt first ϵ messages; here need a global correction

Main takeaway from previous work is that there are good interactive coding schmes

Question in this work: what about blowup in round complexity?

In fact, all prior work assumed one bit per round; also a priori a bound on the communication complexity

Model of This Work

Synchronous message passing model: A and B can send arbitrary length messages; the length of the message can be adaptive; no a priori bound on CC

Given this, this work says good should also include a constant blowup in the total number of rounds; also allow ins+del errors (want to disallow the parties to encode messages via length of the message)

Results

Theorem: exists a good interactive coding scheme; no a priori bound on communication or round complexity; can be adaptive

Also optimize constants so that as errors approaches 0, the round and communication blowup approaches 1

Technical Hurdles

Starting points is BH's backtracking idea; every time you send a message you append a hash of what you think the communication is so far; if an inconsistency then you backtrack (using Schulman's idea of meeting points)

Challengees

1. Need to "smooth out" the protocol; cannot send a long message after a short one
2. Need to erase "inconsistent transcript" carefully; if an inconsistency adversary can cause you to erase an exponentially growing set of messages
3. Hash size depends on entire communication
4. Need to hash the hash
5. Do not know the length of seeds because CC is unknown

2.4 Luca Trevisan on Consensus vs Broadcast, with and w/o Noise

Consensus

Communication network given by an undirected graph

In consensus every node must agree on one of the node's private input

Many different models: e.g. CONGEST and LOCAL

Here look at gossip model

Broadcast

Goal is for one node to transmit its private input to all other nodes

Consensus from Broadcast

If can solve broadcast then can solve consensus; just have all nodes agree on broadcasted value (though have to agree on which node is the node that will broadcast)

For broadcast we have lower bounds but not for consensus

In this paper study if there is a graph between these problems

Gossip Models

Push: node chooses a neighbor to send to

Pull: node chooses a neighbor to receive from

Gossip (push-pull): node chooses a neighbor and can send or receive

Uniform push, uniform pull, uniform gossip: neighbor chosen at random

Consensus and Broadcast in Uniform Gossip Models

Broadcast: $O(\log n)$ in a clique; $\Omega(\log n)$ lower bound by thinking about "infection process"

For consensus $O(\log n)$ in a clique but no super constant lower bounds

Results

First: protocol that transfers lower bounds from broadcast to consensus (for push, pull or general gossip); know $\Omega(\log n)$ rounds to infect all nodes so gives a $\Omega(\log n)$ lower bound

Second: exponential separation in presence of noise; in uniform pull model with noisy single-bit communication; broadcast requires $\Omega(n)$ rounds even in a clique (previous work) but they show consensus doable with $O(\log n)$ rounds

2.5 Yihan Zhang on Generalized List Decoding

L-Packing

Defined ECCs; question is how many balls of radius np can be packed in F_2^n; there is an exponential 1-packing

For list decoding allow overlap but with bounded multiplicity

Allowable p jumps every 2 times; a theorem of Blinovsky gives the dependence

In rest of the talk will generalize this problem

Defined list decoding

2.6 William Lochet on Fault Tolerant Subgraphs with Applications in Kernelization

Given a Digraph, and two subsets S and T

Want to find a subgraph s.t. even after removing k arcs there exists an (s,t) path iff there was a path before

Prior work shows there is always an H with $O(2^k n)$ arcs

Question in this paper: can we reduce dependence on n; not possible in general; e.g. if T is whole graph

But what about when s and t are a single vertex

Lower bounds

If graph is an s-t path then clearly not possible; so some notion of density seems necessary

However, can cheat and adapt this example to be a tournament

Transitive Tournaments

An example where this does work; always a solution of size $k+2$

D is a transitive tournament

Look at ordering of vertices; vertices before s and t are *useless*

If there are fewer than k vertices between s and t then keep all of them

If there are more then there is no k cut

main idea of the paper is to generalize this

look at independence number of the graph; $\mathcal{D}_\alpha$ is all digraphs with fewer than α

Tournaments correspond to $\alpha = 1$

are able to probe a similar result with dependence on α

Definition

A problem $\mathcal{P}$ has a polynomial kernel if instance (U, k) can be turned into an equivalent instance with poly bounds

DFAS

Defined feedback arc set problem

Question: does DFAS has a poly kernel; but this is known for the tournaments case; so want to generalize this to $\mathcal{D}_\alpha$ and so

2.7 Michael Mitzenmacher on Scheduling with Predictions and the Price of Misprediction

Toy Example

Suppose have short and long jobs; if you knew nothing about the jobs you could schedule them just as they come in; but if you knew the job sizes you would put the short jobs first to minimize the avg waiting time

A la ML suppose you had a predictor to guess if job was short or long

So can ask what expected gain is from this alg if predictor has some probability of failure; can work out the details and get some competitive ratio; here not comparing to the perfect algorithm

Takeaway from simplistic model

- Even bad predictions can be helpful

Algorithms with Predictions

A new / growing area; gave some citations

More Complex Model: Single Queues

A more interesting case to consider is in queuing theory

Standard Queueing Model

Poisson arrivals

Service time:

- For theory, exponential services are nice (in practice more heavy-tailed)
- In simple models, service time is unknown

Service discipline:

- FIFO etc; can allow for preemption or not

Main Result for Standard Queues

For M/M/1 queues

- FIFO queue
- Poisson arrivals
- Exponential service times with mean 1

Known Service Times

Can do better than FIFO if have known service times; e.g. shortest job first (not preemptive); shortest remaining processing time (preemptive); preemptive shortest jo b first

Predicted Service Times

What this paper looks at

Suppose an ML algorithm can predict service times; then can replace standard policies with their predicted versions

Obvious Shortcoming for SRPT

If predicted job time is smaller than the actual, eventually the predicted service time will hit 0 and hold the queue until it finishes

So intuition is to prevent big mispredicted job from holding up the queue

Results

Can get a "relatively simple" formula for the price of misprediction in terms of the basic service quantities

Gave some simulation results

2.8 Spyros Angelopoulos on Online Computation with Untrusted Advice

Summary

Models for online computation where algorithm has some limited offline information

Previous work assumes this info is reliable; here study if this does not hold

Online Computation with Advice

Online algorithm receives input in form of a sequence; online alg has to make an irrevocable decision; consider competitive ratio

Situations where alg has access to some offline info; e.g. number of requests

A subfield of online algorithms which deals with questions of this form; a nice survey by Boyar et al.

On The Meaning of Advice

In real world "advice" is a recommendation, not an absolute truth

An example: online ski rental

Described ski rental

Need a model that takes into account untrusted advice bit; i.e. given by an adversary

So have **trusted competitive ratio** and **untrusted competitive ratio**

Objectives

1. Find Pareto-optimal or Pareto-efficient online algs
2. Explore tradeoffs between competitiveness and size of advice

An emerging field

An example: Pareto-Optimality

Let advice be indicator of whether total days is $< B$; can have a hedging parameter and get upper and lower bounds in terms of hedging parameter

More Results

Consider somewhat more technical problems and give more results

E.g. consider online bidding problem; bin packing; list update

2.9 Smoothed Efficient Algorithms and Reductions for Network Coordination Games

Summary of Rsults

Natural dynamics converge in smoothed quasi poly steps

Finding a NE reduces smoothed to local max cut / bisection

Prior Work

Local-max-cut problem is about finding a maximal cut up to flipping one node; but if edge weights are perturbed then w.h.p. any greedy search alg will converge in quasipoly; if graph is complete then even smaller time

Local-max-bisection: find balanced cut maximal up to a swap (no known smoothed analysis)

Smoothed Reductions

Want to map random A input to B case-by-case randomness properties

If these conditions are met, get easier smoothed analysis

Result 1: network coordination games reduce like this to local max cut with 2x2 payoff matrices; with general payoff matrices reduces to local-max-bisection (so smoothed analysis is conditional on LMB)

Common Framework

Main observation is potential function is a linear comb of inputs

Gave a quick sketch of how smoothness-preserving reduction to max cut looks

2.10 Eitan Zlatin on Approximately Strategyproof Tournament Rules: On Large Manipulating Sets and Cover-Consistence

Incentives in sports matter

in 2012 London Badminton game a good team lost early in group stages so they were seeded poorly so other teams tried to lose their games

Question then is what if teams tried to work together to max their chance of jointly winning

What is Tournament

Directed complete graph where direction is who beats whom

A tournament rule maps a tournament to a winner (can be randomized); is **Condorcet-consistent** if whenever a team i that beats everybody then wins for sure

Gave an example where tournament has a cycle; not clear who should win

How Can Teams Manipulate

S subset of teams; two tournaments S-adjacent if they differ only on games where both teams are in S

Defined k-Strongly Non-Manipulable-α tournament rule; α is smallest amount of gain by a coalition of at most k teams

Gave an example of rock-paper-spock where a lower bound

Previous work showed this lower bound is tight; a conjecture for each k the lower bound is tight; this work shows conjecture is false

Gave an example; main tool used is "special" LP

2.11 Jack Wang on Optimal Single-Choice Prophet Inequalities from Samples

Outline

Two cute results on prophet inequality

Will explain background

Present proof of one result

Talk about open questions

What's prophet inequality

Defined prophet inequality

Original problem showed can always get $1/2$ of the optimal value in expectation and this is tight as seen by two boxes, one which is always 1, the other which is $1/\epsilon$ with probability ϵ

Prophet inequalities imply results for mechanism design

A disadvantage of this model is you have to know the exact distributions; would like to do something with data from the past; so replace distributions with samples

Showed can still get $1/2$ while only known 1 sample from each distribution

Moreover, if every box had same distribution previous work showed a tight guarantee of $.745$; here they reduced the number of samples for this result

Proof of First Result

Algorithm: take biggest sample as threshold; take anything that exceeds this threshold

Use **principle of deferred decisions** where real and sample get flipped by a coin; thus only need to prove alg guaranteed over the coin flips

Sort all values (sampled and real values) in descending order; consider first pair of values from same pair

2.12 Robert Robere on Lower Bounds for (Non-monotone) Comparator Circuits

Central open problem (in complexity): lower bounds on relatively weak algorithms; e.g. show SAT requires superlinear size circuits

Defined boolean circuits; crucially gates can fan out their outputs

Boolean circuits are at the boundary of what we don't understand; weaker models of circuits have superlinear bounds; gave known results to this end

Sad fact that with regular boolean circuits only have linear lower bounds (for functions in P)

Goal is to push this boundary

Define comparator circuits that are in between existing LBed circuits and boolean circuits

Comparator Circuits

Has comparator gates: takes two bits and outputs the bits in sorted order; comparator circuit is a circuit only composed of comparator gates

Two natural complexity measures

1. number of wires
2. number of gates

A special case of these are **sorting networks** which have been previously studied; the model for *oblivious sorting algorithms*

Let CC be class of poly computable comparator circuits

CC contains NL, in P; conjecture to be incomparable to NC; complete problem for CC is stable marriage

No superlinear lower bounds known for standard model

Main Result

A superlinear LB on comparator circuits

The function is deciding of all integers input are distinct; any comparator circuit needs $\tilde{\Omega}(n^{3/2})$ wires (stronger than a gate lower bound)

Techniques

Use **Nechiporuk method** but in some sense it can't work

Observation: a circuit for a function also gives a circuit for a restriction of the function

Nechiporuk for Formulas

Choose a partition of inputs $X_1, \ldots, X_t$

Compare the number of subfunctions by restricting function to number of formulas on leaves

Problem is have to bound the number of comparator circuits with W_i wires but can keep adding gates forever; i.e. number of circuits depends on num gates and wires

But as it turns out this claim is false and this will work

Theorem: can keep repeating gates forever if circuit is minimal; w wires means at most w choose 2 wires

So then do Nechiporuk method

2.13 Georg Loho on Signed tropical convexity

Tropical Linear Programming

Slogan: replace $\sum$ by $\max$ and $\cdot$ by $+$

Theorem: checking if a tropical system of equations is in $NP \cap coNP$

Notably, no additive inverses

Checking if a system in a more general setting is NP complete

Motivation

Equivalent to mean payoff games

Parity games form a subclass

Quest for strongly-polynomial time algos

Connection to classic linear programming

Models some scheduling problems

Symmetrized Tropical SemiRing

Use this framework

Idea is similar to defining integers from natural numbers

Double the real numbers and get so called negative tropical numbers but also have to introduce a “third copy”

Then extend max and plus operations

Bad news: no compatible total order for the symmetrized tropical semiring

Signed tropical convex hull

Can define a convex hole in this setting; gets a geometric object with nice properties

E.g. intersection and projection preserve convexity

Get an analogue of Farka’s lemma

Also get an analogue of halfspaces

Open tropical halfpaces are convex but not closed tropical halfspaces

Get back a geometric version of SAT by changing system of tropical equations; shows why this problem is in NP

Also get analogue of Minkowski

Conclusion

Extended tropical convexity for signed tropical numbers

New phenomena (strict vs non-strict inequalities)

Duality and elimination work

New geometric tools

3 January 14, 2020

3.1 Sandeep Silwal on Testing Properties of Multiple Distributions with Few Samples

Motivation

Real world data sets are not identical; e.g. can’t assume a bunch of customers are identical in what they buy but we only have a few data points per person

But also we need these data points to be similar somehow if we want to gain insights

Central question is: how can we model **similarity** (in the context of distribution testing)?

Structural Condition for Similarity

Distribution q (hypothesis)

Sample i from distribution p_i (independently)

Partition domain into two parts A and B such that $\forall p_i$ the probability mass is larger than q on A and smaller than q on B

E.g. if playing slot machines want to distinguish between fair slot machines and not fair slot machines

Results

Known distribution q; ith sample drawn independently from p_i

Can distinguish

- p_i s are identical to q
- p_i s are ϵ far from q in l_1 distance

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

Have to assume that $p_{\rm i}$s assume above structural condition

Natural Question

Is structural condition necessary

In paper show that slightly weaker conditions don't work: e.g. could try there exists some A such that $|p_i(A) - U(A)| \geq \epsilon$ for every i and some set A

Gave example where this weaker condition is satisfied but together these distributions look like the uniform (i.e. collision probability is same and higher order moments match; it's known that if these higher orders match uniform then it's hard to distinguish this distribution from uniform)

Algorithm

Informally

1. Count number of collisions (pairs of samples that match)
2. In the far case, we expect many collisions

Intuition is the birthday paradox: many collisions if far from uniform

Gave formal definition

Glimpse Into Analysis

If $p_1 = p_2 = \ldots = p$ then the prob two samples collide is $||p||^2$

Also if p is far from uniform then get a lower bound on how far 2 norm

Can't do exactly this because all the p_i s are different

It's possible that one pair collides with probability $1/n$ which looks like uniform

But as argued in paper in aggregate these collisions add up

Concluding Remarks

Modeling a heterogeneous dataset is necessary; "finite" deFinetti's theorem doesn't work

On the bright side: maybe existing algorithms are robust to this more general case because in this paper using the standard tools

3.2 Amartya Shankha Biswas on Local Access to Huge Random Objects through Partial Sampling

Huge Random Object

Consider random walk that goes up and down with probability $1/2$

What if you only care about a few positions? Seems inefficient to produce all walk positions if just want to know height at time t

Only restriction is queries need to be consistent with some random walk

Query Requirements

Formally, want efficiently (poly log)time, space, random bits per query

Want output distribution to be ϵ close to true distribution

Example Queries in Random Graphs

E.g.

- Check if an edge is present
- Check all neighbors of a vertex or next neighbor in adjacency list
- Take a random-neighbor of a vertex

Gave some prior work that e.g. introduced the model which focused on pseudo-random limited queries; sparse version of this and preferential attachment version of this

Results for $G(n, p)$ Graphs

Defined $G(n, p)$

Will focus on single row of adjacency matrix

Next Neighbor and Random-Neighbor

Consider case where $p < 1/\text{poly} \log n$ (i.e. dense graph)

Here, could just flip coins left to right until you see a 1 for next-neighbor

If graph is sparse can adapt previous work above

So interesting case is intermediate p; e.g. nodes have $\sqrt{n}$ neighbors

Skip-Sampling

For next neighbor could skip ahead since this is just geometric distribution (if nothing ahead has yet been determined); but problem is you might have known entries interspersed with unknown entries

Dealing With Known Entries

For known entries when these are discovered they are reported; but not the 0 s

First step is to ignore known entries until you fill whatever range you're trying to fill; but then you have conflicts; to resolve these just inherit the old 1s; but to inherit the 0s you have to check your own new 1 entries (so runtime is bounded by number of your new 1 entries)

This is next-neighbor; one more step needed for random-neighbor; here partition into buckets with expected $O(\log n)$ ones in each bucket; when fill out, fill out entire bucket; have to also do some rejection sampling

Open Questions

Degree queries etc.

Some other Results

Also extend to other random models and some random walks like Dyck paths

Results on random colorings of huge graphs

3.3 Arsen Vasilyan on Monotone probability distributions over the Boolean cube can be learned with sublinear samples

Learning Probability Distributions

Some unknown distribution; alg receives samples from ρ; needs to output $\hat{\rho}$; want small variation distance

Need $\Omega(N)$ samples where N is domain size if you don't assume anything about distribution; so should assume something

Defined Boolean cube; break cube into "levels"; i.e. sets with same hamming weight; a standard partial order by containment of corresponding sets

ρ is **monotone** if x greater than y means that x has greater probability

Main Result

If ρ is monotone can learn with $2^n/2^{\Theta(n^{1/5})}$; notably this is sublinear

Outline

1. Main lemma
2. Lemma gives algo
3. Open problems

Definitions

Say x is **tight** for monotone if x is it's largest predecessor; otherwise **slacky**

Gave example of tight and slacky

Precursor to Main Lemma

Prior work showed a monotone Boolean function can be broken into "structured" part and "noise" part where noise is small and always positive and structured part is slacky and x is in a constant number of levels

Idea is to take this lemma and prove it for distributions but unfortunately this doesn't work; have to fix this by introducing a **weight** to each level; bound total weight of special levels instead of the number as in the prior work; then lemma is true

How Lemma Leads to Algorithm

Between special levels boundaries are slacky and in the middle is tight; get $\rho(x) = \rho(y)$ where y is largest predecessor of x in the middle which is also just the average of points between x and y; so estimate $\rho(x)$ by estimating this average

Many issues still to overcome: like where is y; where are special levels

The Algorithm

Computes this above empirical average

Open Problems

3.4 David Mass on Local-to-Global Agreement Expansion via The Variance Method

Defined agreement expander / local agreement

Defined link of a vertex in a simplicial complex: the simplicial complex induced by looking at one vertex (and removing it)

Main Result

If all links of HDX are agreement expanders then the whole complex is an agreement expander

Proof by variance method; a tool they develop

The Variance Method

The variance of a function on sets of size d shrinks by a factor of d when restricted to vertices

Gave "local-to-global" proof idea

Summary

Interested in which sparse set systems are agreement expanders?

Known constructions of HDXs don't meet previous requirements

Their work shows that agreement expansion in the links gives agreement expansion of the whole complex; applies to Ramanujan complexes

Proof by variance method

3.5 Yael Hitron on Random Sketching, Clustering, and Short-Term Memory in Spiking Neural Networks

Consider algorithmic aspects of biological networks but from distributed perspective; in **spiking neurological model**

Consider tasks of and compression and clustering

Demonstrated networks that solved these tasks

Spiking Neural Networks

A digraph with weights on edges; two types of neurons: excitatory and inhibitory

Should think of neuron as probabilistic threshold

The potential of a neuron is the weight of incoming weights minus some bias; fires according to sigmoid of this potential

Network proceeds in discrete synchronous rounds

Computational problems in SNN: given special input neurons X and output neurons Y and a target function f; complexity is size and rounds needed by network

Problems

Neural Clustering Problem: n input neurons; k output neurons where $k \leq n$; want to map similar things to similar things where similar is hamming distance

Motivated by fruit fly olfactory system

Main Result

There is a network solving this problem with probability $1-\epsilon$ using $\tilde{O}(1/\Delta^{3/2})$ auxiliary neurons in $\tilde{O}(1/\Delta)$ time where Δ is bound on hamming distance

High level: network has 3 steps; first random project to smaller set; then map to sparse vector set with sparsification set; finally take sparse vectors and map to unique representative in a "sequential mapping"

Step 1

Connect two layers with complete bipartite graph where weights drawn from χ^2 distribution (because it's non-negative)

This layer has with good prob that the max firing rate is different for different patterns of different neurons

Step 2

Take intermediate neurons and convert to a vector of same size where only firing neuron is the one with max firing rate;

Step 3

Use "association" neurons that fire if at least one of their corresponding neurons; also have "memory" mechanism that inhibits

Other Results

Can modify construction to map similar inputs to same output (clustering)

Also implemented some sort of short term memory

Also modified to implement a biological **bloom-filter**

3.6 Gal Sadeh on Sample Complexity Bounds for Influence Maximization

Intuition

Given big social network; you're a user and post something; want to understand how post spreads through network / your impact re people who read your post

The influence max problem is to find a group of people that have max influence on the network

Many diffusion models for how information spreads throughout the network

One such model is Independent Cascade

IC Model

Each edges has a probability, generate samples by taking edges according to their probabilities (independently)

Gave example

A generalization of this is the b-dependence model

b-Dependence Model

Have sets of edges which with some probability and now take all edges in a set according to their probability; if $b=1$ then just the IC model

Diffusion in IC Model

For a given sample, start with seed set; and activate them and their neighbors up to τ steps

The **influence** is the expectation of the size of these reachability sets

A further generalization

Stochastic Diffusion Models (SDM)

A distribution over activation functions where a node is activated as a function of whether or not its neighbors are activated

Generalized influence: now not the number of nodes reached but just some arbitrary utility function applied to the set of reached nodes

Independent Strongly Submodular SDMs

Acitvation functions drawn indep

Generalized influnce utility function is a monotone submodular function

One other property

Influence maximization problem now is to find small set s with largest influence

The sample complexity is the number of samples needed to solve this problem

Results

Gave main result on sample complexity where now have dependence on τ instead of n

Gave various bounds for various models

Computational Efficiency

Until now just talked about sample complexity

Problem is NP-hard to approximate better than $1 - 1/e$; greedy algorithm gets essentially $1 - 1/e$

Main contribution here is an implementation for greedy with small sample complexity

A better algorithm for "low variance" models

3.7 Alex Wein on Computational Hardness of Certifying Bounds on Constrained PCA Problems

Part 1

Computational hardness for statistical problems

E.g. in planted clique have a random graph with a k-clique chosen at random; goal is to find the clique

Statistically, can find the clique as long as it's larger than $2 \log n$

But in poly time can only find $\Omega(\sqrt{n})$ clique

So statistically it's possible, but not computationally

This phenomena occurs in many other problems in high-dimensional signal + noise problems

Question is how to show these problems are hard

- Reductions from planted clique
- Properties of solution space / failure of certain algorithms

- Sums of squares lower bounds
- This talk: "low-degree" method as proposed by SoS people

Low-Degree Method

Suppose want to hypothesis test between two distributions e.g.

- null model vs
- planted model

Key idea: ask if there is a low degree pol that distinguishes these two: want poly to be big on planted and small on null model

So look at ratio of expectations of this polynomial on two settings where max over all polys; surprisingly you can compute this

This centers on **low degree conjecture**: polys of degree $O(\log n)$ are as powerful as all poly time algorithms; so if can show this ratio is bounded can argue there are no poly time algos

Evidence for Conjecture

1. Holds for many problems
2. If degree $\log n$ fail so too do **spectral methods**

Advantages of Low-Degree Method

Much simpler than SoS lower bounds

By varying degree can explore power of subexponential-time algorithms

Interpretable reason for what makes some problems easy/hard

Part 2: Hardness of Certification for Constrained PCA Problems

Constrained PCA

W is a random matrix of Gaussians

Eigenvalues follow a semicircle on $[-2, 2]$

PCA will tell you to find max eigenvalue (2 whp)

Contrained PCA: same but must max over hypercube

Search vs Certification

Two computational problems for this problem:

- Search: find x that achieves large $x^T W x$
- Certification: determine if OPT at most B

Search and certification can be thought of as proving a lower and upper bound respectively

Prior Work

Perfect search possibly in poly time

Trivial spectral certification: compute largest eigenvalue and ignore hypercube constraint

Question is can we do better certification in poly time

Natural strategy is

- convex relaxation; e.g. SoS

Main Result

Main result: But actually cannot do better than trivial certification given low-degree conjecture

In fact need essentially time $2^{n^{1-o(1)}}$ time assuming low-degree-conjecture

Proof Outline

1. Reduce from hypothesis testing problem to certification problem
2. Use low degree method

Spiked Subspace

Problem that reduce from

Null model: random subspace

Planted model: a planted hypercube vector in the subspace

Low-degree method suggests exp time needed to do this

Other half of the theorem reduces to certification

Summary

Low-degree method is systematic way of predicting when hypothesis testing is easy/hard

For constrained PCA gave low-degree eveidence that non-trivial certification is hard

3.8 Shuichi Hirahara on Unexpected Power of Random Strings

Randomness

Complexity: BPP (defined)

Computability: R_{K_U} Kolmogorov random strings (not computable)

A conjecture that tries to understand former via latter

Allender's Conjecture

Conjecture: BPP is set of all R_{K_U} strings

Result today: this conjecture is false under standard complexity assumptions

Outlines

1. Kolmogorov-randomness
2. Allender's Conjecture
3. Results

Kolmogorov Complexity

Defined KC (shortest program size to print x)

To formally define need to fix an interpreter U where U is interpreter of program

Choose U so KC is smallest; a machine U is **universal** if the KC with this U is at most the KC under every other interpreter up to an additive $O(1)$

Take such a U

Kolmogorov Randomness

A string x is random if $K_U(x) \geq |x|$

Let R_{K_U} be all random strings

Allender's Conjecture

Set R_{K_U} is not computable

Main question: What can be solved efficiently by nonadaptively asking whether $q \in R_{K_U}$ (no matter what U)

Let $P_{||}^{R_{K_U}}$ be all languages solvable with this.

Known that $BPP \subseteq P_{||}^{R_{K_U}} \subseteq PSPACE$

So Alender's conjecture is $P_{||}^{R_{K_U}} = \bigcap_U P_{||}^{R_{K_U}}$

Motivation: $\text{MCSP}^{\text{HALT}} \approx R_{K_U}$; several researchers tried to verify this

Intuition of AC

Gave intuition for why would think AC is true

Results

Can consider exponential version of classic complexity classes

Main theorem: shows $EXPH \subseteq PH^{R_{K_U}}$

Then can use a standard padding argument to refute Allender's conjecture; explained padding argument; e.g. $NP \subseteq P$ implies $NEXP \subseteq EXP$

Proof Sketch

Replace exponentially-long certificates with poly size circuits using R_{K_U} oracle

3.9 Ninad Rajgopal on Beyond Natural Proofs: Hardness Magnification and Locality

Defined boolean circuits (DAG; size is num gates; P/Poly) and formulas (bin tree; $O(\log n) depth$ size is leaves; NC^1)

Natural Proofs Barrier

Circuit lower bounds are hard to prove; some evidence of why lower bounds are hard to show is natural proofs by Rudich

There exists no P/Poly-natural proofs useful against P/Poly; i.e. if want lower bounds for P/Poly they cannot be naturalizable

Hardness Magnification

Candidate for showing lower bounds; if show barely super-linear lower bounds then this implies a strong lower bound against P/Poly or NC^1

E.g. $n^{1.1}$ shows $NP \not\subseteq NC^1$

Minimum circuit size problem (MCSP): given a truth table want to decide if function computable by circuits of size $\leq s$; don't know if this is NP-hard or hard to approximate

This Paper

More examples of HM

Something else

Question 1: Does HM avoid natural proofs barrier; in a certain sense yes if start from *approx*-MCSP

Theorem: If MCSP not in size $N^{1.01}$ then no P/poly-natural properties against P/Poly

HM Frontiers

What makes magnification promising?

Question 2: adapt Tal's lower bound proofs to show lower bound required for magnification?

3.10 Karthik C. S. on Hardness Amplification of Optimization Problems

Guiding motivation: complexity people care about circuits; algos people care about graphs and strings; let's build a bridge

Want to build a theory of average case complexity; also the cornerstone of modern crypto (uses hard on average functions)

Modest Goal: hardness amplification; start from average case and raise it to sharp average case

Dream Theorem in Hardness Amplification

Suppose a family of functions (graphs)

Suppose every algo fails on certain fraction ($p(n)$) of these inputs in time $t(n)$

Want to show this means there's a different family such that every algo will fail on some much larger fraction of inputs (i.e. amplified failure probability)

This has been proven for something like the permanent and EXP

What about NP? Can do something like this when working with circuits; can go from failing on $1/\text{poly}(n)$ to failure probability of $1/2$; can be made to work with algorithms

Big open arenas in hardness amplification: optimization problems

Optimization Problems

This work: A general technique to do hardness amplification for optimization problems

- Many NP-hard problems
- Subquadratic-hard problems
- Total problems

Will give a flavor of how these things

Max Clique

Defined problem

Show given a distribution over graphs on n vertices; if any randomized algorithm will fail with probability $1/n$ can get a new distribution such that algorithm will fail on .99 fraction

Proof Overview

Idea is to define algo for original distribution that succeeds with too high probability with new distribution; a contradiction

1. Independently sample from original dist
2. H is disjoint union of k graphs
3. Insert all edges across graphs
4. Output H

So algorithm just runs new algo on old one but with graph you're interested in planted in the ith position chosen at random; project the solution onto original graph

Use "Feige-Kilian" direct product theorem

3.11 Rahul Santhanam on Pseudorandomness and the Minimum Circuit Size Problem

Many results; will focus on one

<u>One-Way Functions</u>

A function that's easy to compute but hard to invert in average case; i.e. for any random x, any poly-time algorithm succeeds with negligible probability

Funcamental in pseudorandomness (equivalent to pseudorandom generators and pseudorandom function generators) and crypto (imply symmetric-key encryption, message authentication etc.)

But re complexity theory, it's not well understood how these relate to traditional complexity notions

Ideally want to base on-way functions on $NP \neq P$; however evidence this cannot be done via "black box" reductions

In this work suggest a candidate to do this relation: the minimum circuit size problem (MCSP); will make some partial progress towards relating hardness of MCSP to OWFs

Defined MCSP

Longstanding question: is MCSP NP-complete (clearly in NP)

<u>Average-Case Complexity</u>

A natural distribution on MCSP; namely uniform over Boolean functions

However, trivial to solve on average with errors so instead consider a zero-error notion of average-case complexity; i.e. circuits which are always correct or output "?"

Proposition: MCSP is average-case hard iff natural proofs against sub-exponential size circuits do not exist

<u>This work</u>

In some ways this talk is a failure about answering this question

To show result need a **universality conjecture**: a fixed function that is a PRG whose range consists of strings with non-trivial circuit complexity if there is *any* pseudorandom dist. . .

<u>Main Result</u>

Assuming this conjecture TFAE

- MCSP hard on average
- OWFs exist
- Pseudorandom generators with poly stretch exist
- Natural proofs against sub-exp size circuits don't exist

- Poly size circuits cannot be PAC-learned over uniform distribution in poly time

Impagliazzo's Five Worlds

Stated 5 worlds:

1. Algorithmica ($P = NP$)
2. Heuristica (NP is avg case hard)
3. Pessiland (OWFs exist)
4. Minicrypt (Public-key crtpyo)
5. Cryptomania

Assuming universality allows this work to rule out heuristica and pessiland

3.12 Guy Blanc and Jane Lange on Top-down induction of decision trees: rigorous guarantees and inherent limitations

This work: learning decision trees from labeled data

Goal is to output a decision tree not much larger than the underlying decision tree

This is useful in practice

Decision trees are also intensively studied in TCS

- Define query model of computation
- Applications in quantum complexity
- Derandomization
- Learning theory

A disconnect between theory and practice of learning decision trees

- In practice build decision tree from top down
- Algs with strong theoretical guarantees work from bottom up

This work bridges this disconnect

Part 1: guarantees for top down algs

Alg:

1. determine "good" variable to query as root
2. recurse on subtrees

"good" is like relevant or important etc

Formalize this with notion of influence; influence of a variable is probability that over random input flipping this variable flips the function output

Let "good" be most influential variable

Result: give probable guarantees for this

Give a guarantee for this alg; essentially quasi polynomial

Also give a matching lower bound for this algorithm

A tighter upper bound for monotone functions; also give another lower bound for monotone; here a gap

Algorithmic Consequences

Can properly learn decision trees in time about $s^{O(\log s)}$

Downside is require query access to the function (to compute variable influence)

Get better runtime for monotone functions ($s^{\sqrt{\log s}}$) and also show that the criteria of existing algos used in practice is identical to choosing max influence

Part 2: theoretical algorithms with improved guarantees

Improve the space use and sample complexity of existing algorithm for properly learning decision trees; here deviate by not forcing their algorithm to exactly match the sample data allowing them to limit the depth of their trees

3.13 Domagoj Bradac on Robust Algorithms for the Secretary Problem

Classic Secretary

n items; each arrives at times uniformly at random in $[0, 1]$; must decide to accept or reject; choices are immediate or irrevocable

In value maximization: values are chosen adversarially but arrival times are random

In probability maximization: items have only a relative order but want to max probability of max item

Gave an example instance

Classic theorem of Dynkin: can get largest item with probability at least $1/4$ (first observe first half and then use largest in first half as threshold)

The $1/4$ can be improved to $1/e$ which is best possible

However, most algs in this setting are fragile (arrival times are uniform and independent)

Robust Secretary

Natural question is do we need random arrivals

If fully adversarial can't beat random guessing (i.e. $1/n$)

Thus, consider mixed arrival times in this paper

- green honest random arrival times
- red items arrive adversarially (and times chosen before green items)
- OPT is the 2nd green max (because green max is unattainable)

Theorems of this work:

- value max: get $1/O(\log^* n)^2$ approx
- probability maximization: get $\geq 1/O(\log^2 n)$ approx
- choosing multiple items: get $(1 - \epsilon)$ approx provided lower bound on density of objects

Flavor of Techniques

Consider single item value max

Shows how to get poly $\log n$ approximation w.p. $1/3$ pick a random item

Then partition $[0, 1]$ into two halves

w.p. 1/3 run Dynkin in 1st half

w.p. observe the first half and let a be max value (know OPT $\in [a/n, a]$); partition this interval into buckets, pick a random bucket and take first element with value in this bucket

Then showed how to improve to $O(\log^* n)^c$ approximation

Open Problems

A superconstant lower bound in the single item setting

Can the probability max algorithm be made constructive?

Extend results to general packing LPs

3.14 Daogao Liu on Algorithms and Adaptivity Gaps for Stochastic k-TSP

Classic k-TSP

Given metric and target value k; want tour of at least k points while minimizing length (also don't need to go back to start point)

a 2-approximation by garg

stochastic k-tsp

two version considered; will only introduce one (stoch-reward k) tsp

demand is random; sell at least k brushes but know distributions of demands

want to minimize the expected length until enough reward collected

actual reward revealed once visited; no reward for revisiting

adaptivity gap

focus of paper

adaptive: sequence may depend on the instantiation of the random rewards

non-adaptive: visited points fixed beforehand

adaptivity gap is ratio of non-adaptive opt to adaptive opt

approximation ratio is ratio of alg to adaptive opt

prior work

an adaptive $o(\log k)$

a non-adaptive $o(\log^2 k)$

adaptivity gap is $o(\log^2 k)$

main question is are there $o(1)$

this work

this work gives non-adaptive $o(1)$ approximation

key task

so what does alg do; transfer original problem into a "key task"

a key task is stoch-reward instance along with a ravelling budget b

objective:

- any adaptive alg a with budget b
- design non-adaptive a' with budget $200b$ s.t. a' gets more reward than a

Bernoulli case

a special case where distributions are Bernoulli

want to find tour in remaining graph within budget b via orienteering

general case

distributions are arbitrary

problem is "expectation may not be a good indicator"

find a way to "compute a critical j" and then truncate distribution by j so that expectation is a "good indicator"

open problem

if ratio can be improved to small constant like 2

3.15 Ruben Becker on low diameter graph decompositions by approximate distance computation

Talk about graph decompositions

motivation

for many graph problems: useful to decompose problem; especially in models of large-scale computation

useful properties of decomposition:

- parts have small diameter
- edges unlikely to be cut
- short edges less likely to be cut than long edges

(this is an ldd)

LDDs

(d, λ)-decomposition is a partition of v into clusters c s.t.

- diam of cluster at most d
- prob edge cut at most $\lambda \cdot l_e/d$ where l_e is length

There are **strong** and **weak** diameter versions of these

Tree-Supported Decomposition

Notion introduced in this paper which is between strong and weak; each cluster has a support tree T_C spanning it but may contain nodes not in it

Want num trees an edge in is to be small

Contributions

Give construction like before where load is at most $O(\log n)$ but whereas old LDDS need exact SSSP algos, these need only approx SSSP algos

Crucial ingredient: “blurry ball growing”

Blurry Ball Growing

Described ball growing and showed why this approach fails if the shortest paths are approximate: nodes at small distances are still fine but if they are at large distances it’s bad

Idea is to start with standard ball and then “blur” it’s boundary: I.e. sample ball radius and then extend the ball by randomly sampled value

Use idea of “safe” edge: i.e. one that is sufficiently far to not be cut or is already inside a ball; show probability that edge never becomes safe can be upper bounded

Tree-Supported Decomposition

Now use this blurry ball growing to get above decomposition; but need to do it in parallel a la MPX since interested in distributed/parallel applications

So what do is remove boundary of partition, only keep the interior of each cluster; use blurry ball growing to fix the boundaries so that don’t grow into each other

Then remove obtained blurred clusters and recurse; only get $O(\log n)$ depth

HSTs

Defined HSTs

Now to get HSTs with this use the decomposition tree of hierarchical TSD

Conclusion

Message to remember: LDDs can be computed with just approx shortest paths

3.16 Jayson Lynch and Dylan Hendrickson on Toward a General Complexity Theory of Motion Planning: Characterizing Which Gadgets Make Games Hard

A framework for proving motion-planning problems are hard

Have a graph with some gadgets in it; e.g. the locking 2-toggle where when you move across the gadget the state of the gadget gets changed so you can only go back through the side you went through (and toggles back if you return through it)

Results

Decision problem: can you reach a goal in some network

Also consider variants

- 2 players racing to goal
- Team game where multiple agents racing and imperfect information
- Also axis of polynomially bounded and polynomially unbounded

Reversible Deterministic Gadgets

When cross gadget can undo and it’s deterministic; e.g. locking 2-toggle

Characterizing these

1. non-interacting gadgets are in NL
2. locking 2-toggle is PSPACE complete etc; reduction from non-deterministic constraint logic

Non-interacting tunnels

Two tunnels are interacting if a transition along one changes dynamics of another

Show everything simulates locking 2-toggle; so any of these gadgets is PSPACE complete

For planar case many more gadgets where show all gadgets simulate one another

2 Player Games

Non-interacting tunnels not easy when 2 competing players

Team Game

Undecidable for interacting reversible gadgets; surprising because a polynomially-sized game

DAG Gadgets

Look at family of DAG gadgets for 1-player games

Show NP completeness via SAT

For 2-player versions show PSPACE complete from QBF

For team version show NEXP complete

3.17 John Sylvester on Choice and Bias in Random Walks

A Tale of Three Walks

Focus on cover time: time to visit every vertex in the graph

Main characters

- simple random walk (SRW); usual random walk
- choice random walk (CRW): rather than moving to random neighbor you're offered two and must choose one; the "why" is to minimize cover time; the "how" is player has any amount of resources they want; inspired by power of two choices
- ϵ-biased random walk (ϵ-BRW): at every step flip a coin where if fail then move to random neighbor but if succeed get to choose

Mostly interested in CRW; the ϵ-BRW defined in prior work where looked at hitting time and stationary probabilities

Results

Can simulate ϵ-BRW with CRW provided ϵ not too big

Theorem proved: basically shows can make an improvement over simple random walk for hitting and cover time for ϵ-BRW and CRW

Conjecture of Azar et al.

Was conjectured that can increase probability on a vertex by ϵ-BRW

Was previously proved for bounded degree

This work proved it for regularish and something else graph

Main Tool

Can “boost” transition probabilities (like prior conjecture but now with transitions instead of stationary distribution)

Encode all walks from origin as vertices in a tree of height t

Complexity

Also interested in complexity of finding optimal strategies

They adapted LP of Azar et al. for CRW to max/min a stationary probability

New: showed minimizing cover time is NP-hard for CRW and ϵ-BRW