THREE PUZZLES ON MATHEMATICS, COMPUTATION, AND GAMES

GIL KALAI
HEBREW UNIVERSITY OF JERUSALEM AND YALE UNIVERSITY

ABSTRACT. The second puzzle is about errors made when votes are counted during elections.

1. INTRODUCTION

The theory of computing and computer science as a whole are precious resources for mathematicians. They bring new questions, new profound ideas, and new perspectives on classical mathematical objects, and serve as new areas for applications of mathematics and of mathematical reasoning. In my lecture I will talk about three mathematical puzzles involving mathematics and computation (and, at times, other fields) that have preoccupied me over the years. The connection between mathematics and computing is especially strong in my field of combinatorics, and I believe that being able to personally experience the scientific developments described here over the last three decades may give my description some added value. For all three puzzles I will try to describe with some detail both the large picture at hand, and zoom in on topics related to my own work.

Puzzle 2: What are methods of election that are immune to errors in the counting of votes? The second puzzle can be seen in the context of understanding and planning of electoral methods. We all remember the sight of vote recount in Florida in the 2000 US presidential election. Is the American electoral system, based on electoral votes, inherently more susceptible to mistakes than the majority system? And what is the most stable method? Together with Itai Benjamini and Oded Schramm we investigated these and similar problems. We asked the following question: given that there are two

Work supported in part by ERC advanced grant 320924, BSF grant 2006066, and NSF grant DMS-1300120.

FIGURE 1. Recounts in the 2000 elections (drawing: Neta Kalai).

candidates and each voter chooses at random and with equal probability (independently) between them, what is the stability of the outcome, when in the vote-counting process one percent of the votes is counted incorrectly? (The mathematical jargon for these errors is "noise.") We defined a measure of noise sensitivity of electoral methods and found that weighted majority methods are immune to noise, namely, when the probability of error is small, the chances that the election outcome will be affected diminish. We also showed that every stable–to–noise method is "close" (in some mathematical sense) to a weighted majority method. In later work, O'Donnell, Oleszkiewicz, and Mossel showed that the majority system is most stable to noise among all non-dictatorial methods.

Our work was published in 1999, a year before the question appeared in the headlines in the US presidential election, and it did not even deal with the subject of elections. We were interested in understanding the problem of planar percolation, a mathematical model derived from statistical physics. In our article we showed that if we adopt an electoral system based on the model of percolation, this method will be very sensitive to noise. This insight is of no use at all in planning good electoral methods, but it makes it possible to understand interesting phenomena in the study of percolation.

After the US presidential election in 2000 we tried to understand the relevance of our model and the concepts of stability and noise in real-life elections: is the measure for noise stability that we proposed relevant, even though the basic assumption that each voter randomly votes with equal probability for one of the candidates is far from realistic? The attempt to link mathematical models to questions about elections (and, more generally, to social science) is fascinating and complicated, and a true pioneer in this study was the Marquis de Condorcet, a mathematician and philosopher, a democrat, a human rights advocate, and a feminist who lived in France in the 18th century. One of Condorcet's findings, often referred to as Condorcet's paradox, is that when there are three candidates, the majority rule can sometimes lead to cyclic outcomes, and it turns out that the probability for cyclic outcomes depends on the stability to noise of the voting system. In Section 2 we will discuss noise stability and sensitivity, and various connections to elections, percolation, games, and computational complexity.

2. Elections and noise

2.1. **Games 2: Questions about voting games and social welfare.**

2.1.1. *Cooperative games.* A cooperative game (without side payments) is described by a set of n players N. and a payoff function v which associates to every subset S (called *coalition*) of N a non-negative real number $v(S)$. Cooperative games were introduced by von Neumann and Morgenstern. A game is monotone if $v(T) \geq v(S)$ when $S \subset T$. We will assume that $v(\emptyset) = 0$. A voting game is a monotone cooperative game in which $v(S) \in \{0, 1\}$. If $v(S) = 1$ we call S a winning coalition and if $v(S) = 0$ then S is a losing coalition. Voting games represent voting rules for two-candidate elections, the candidate being Anna and Bianca. Anna wins if the set of voters that voted for her is a winning coalition. Important voting rules are the majority rule, where n is odd and the winning coalitions are those with more than $n/2$ voters, and the dictatorship rule, where the winning coalitions are those containing a fixed voter called "the dictator." Voting games are also referred to as monotone Boolean functions.

2.1.2. *How to measure power?* There are two related measures of power for voting games and both are defined in terms of general cooperative games. The Banzhaf measure of power for player i (Also called the influence of i) is the expected value of $v(S \cup \{i\}) - v(S)$ taken over all coalitions S that do not contain i. The Shapley value of player i is defined as follows: For a random ordering of the players consider the coalition S of players who come before i in the ordering. The Shapley value $s_i(v)$ is the expectation over all $n!$ orderings of $v(S \cup \{i\}) - v(S)$. (For voting games, the Shapley value is also called the Shapley-Shubik power index.) For voting games if $v(S) = 0$ and $v(S \cup \{i\}) = 1$, we call voter i *pivotal* with respect to S.

2.1.3. *Aggregation of information.* For a voting game v and $p, 0 \le p \le 1$ denote by $\mu_p(v)$ the probability that a random set S of players is a winning coalition when for every player v the probability that $v \in S$ is p, independently for all players. Condorcet's Jury theorem asserts that when $p > 1/2$, for the sequence v_n of majority games on n players $\lim_{n \to \infty} \mu_p(v_n) = 1$. This result, a direct consequence of the law of large numbers, is referred to as asymptotically complete aggregation of information.

A voting game is *strong* (also called *neutral*) if the complement of a winning coalition is a losing coalition. A voting game is strongly balanced if precisely half of the coalitions are winning and it is balanced if $0.1 \le \mu_p(v) \le 0.9$. A voting game is weakly symmetric if it is invariant under a transitive group of permutations of the voters.

Theorem 2.1 (Friedgut and Kalai (1996), Kalai (2004))**.** *(i) Weakly-symmetric balanced voting games aggregate information.*

(ii) Balanced voting games aggregate information iff their maximum Shapley value tend to zero.

2.1.4. *Friedgut's Junta theorem.* The total influence, $I(v)$, of a balanced voting game is the sum of Banzhaf power indices for all players. (Note that the sum of Shapley values of all players is one.) For the majority rule the total influence is the maximum over all voting games and $I = \theta(\sqrt{n})$. The total influence for dictatorship is one which is the minimum for strongly balanced games. A voting game is a C-Junta if there is a a set J, $|S| \le C$ such that $v(S)$ depends only on $S \cap J$.

Theorem 2.2 (Friedgut's Junta theorem (1998))**.** *For every $b, \epsilon > 0$ there is $C = C(b, \epsilon)$ with the following property: For every $\epsilon, b > 0$ a voting game v with total influence at most b is ϵ-close to a C-junta g. (Here, ϵ-close means that for all but a fraction ϵ of sets S, $v(S) = g(S)$.*

2.1.5. *Sensitivity to noise.* Let $w_1, w_2, \dots, w_n$ be nonnegative real weights and T be a real number. A weighted majority is defined by $v(S) = 1$ iff $\sum_{i \in S} w_i \ge T$.

Consider a two-candidate election based on a voting game v where each voter votes for one of the two candidates at random, with probability 1/2, and these probabilities are independent. Let S be the set of voters voting for Anna, who wins the election if $v(S) = 1$. Next consider a scenario where in the vote counting process there is a small probability t for a mistake where the vote is miscounted, and assume that these mistakes are also statistically independent. The set of voters believed to vote for Anna after the counting is T. Define $N_t(v)$ as the probability that $v(T) \ne v(S)$. A family of voting games is called uniformly noise stable if for every $\epsilon > 0$ there exists $t > 0$ such that $N_t(v) < \epsilon$. A sequence v_n of strong voting games is noise sensitive if for every $t > 0$ $\lim_{n \to \infty} N_t(v_n) = 1/2$.

Theorem 2.3 (Benjamini, Kalai, Schramm (1999))**.** *For a sequence of balanced voting game v_n each of the following two conditions imply that v_n is noise sensitive:*

(i) The maximum correlation between v_n and a balanced weighted majority game tends to 0.

(ii) $\lim_{n \to \infty} \sum_i b_i^2(f) = 0$.

2.1.6. *Majority is stablest.*

Theorem 2.4 (Sheppard (1899))**.** *Let v_n be the majority voting games with n players.*

$$\lim_{n \to \infty} N_t(v_n) = \frac{\arccos(1-t)}{\pi}.$$

Theorem 2.5 (Mossel, O'Donnell, and Oleszkiewicz (2010))**.** *Let v_n be a sequence of games with diminishing maximal Banzahf power index. Then*

$$N_t(v_n) \leq \frac{\arccos(1-t)}{\pi} - o(1).$$

2.1.7. *The influence of malicious counting errors.* For every balanced voting games

Let S be a set of voters. $I_S(v)$ is the probability over sets of voters T which are disjoint from S that $v(S \cup T) = 1$ and $v(T) = 0$.

Theorem 2.6 (Kahn, Kalai, Linial (1988))**.** *For every balanced monotone voting game v.*

(i) There exists a voter k such that

$$b_k(f) \geq C \log n/n.$$

(ii) There exists a set S of $a(n) \cdot n/\log n$ voters, where $a(n)$ tends to infinity with n as slowly as we wish, such that $I_S(f) = 1 - o(1)$

This result was conjectured by Ben-Or and Linial (1985) who gave a "tribe" example showing that both parts of the theorem are sharp. Ajtai and Linial (1993) found a voting game where no set of $o(n/\log^2(n))$ can influence the outcome of the elections in favor of even one of the candidates.

2.1.8. *"It ain't over 'till it's over" theorem.* Consider the majority voting game when the number of voters tends to infinity and every voter votes for each candidate with equal probability, independently. There exists (tiny) $\delta > 0$ such that when you count 99 percents of votes chosen at random and Anna is ahead in the count, still with probability tending to one, the probability that the remaining votes will change the picture is larger than δ. We refer to this property of the majority function as the (IAOUIO)-property. Clearly, dictatorship and Juntas do not have the (IAOUIO)-property.

Theorem 2.7 (Mossel, O'Donnell, and Oleszkiewicz (2010))**.** *Every sequence of voting games with diminishing maximal Banzhaf power index has the (IAOUIO)-property.*

2.1.9. *Condorcet's paradox and Arrow's theorem.* A generalized social welfare function is a map from n voters' order relations on m alternatives, to an antisymmetric relation for the society, satisfying the following two properties.

(1) If every voter prefers a to b then so is the society. (We do not need to assume that this dependence is monotone.)

(2) Society's preference between a and b depends only on the individual preferences between these candidates.

A social welfare function is a generalized welfare function such that for every n-tuples of order relations of the voters, the society preferences are acyclic.

Theorem 2.8 (Arrow)**.** *For three or more alternatives, the only social welfare functions are dictatorial.*

Theorem 2.9 (Kalai (2002), Mossel (2012), Keller (2012))**.** *For three or more alternatives the only nearly rational social welfare functions are nearly dictatorial*

Theorem 2.10 (Mossel, O'Donnell, and Oleszkiewicz (2010))**.** *The majority gives asymptotically "most rational" social preferences among social welfare functions based on strong voting games with vanishing maximal Banzhaf power.*

A choice function is a rule which based on individual ranking of the candidates, gives the winner of the election. Manipulation (also called "non-naive voting" and "strategic voting") is a situation where given the preferences of other voters, a voter may gain by not being truthful about his preferences.

Theorem 2.11 (Gibbard (1977), Satterthwaite (1975))**.** *Every non dictatorial choice function is manipulable.*

Theorem 2.12 (Friedgut, Kalai, Keller, Nisan (2011), Isaksson, Kindler, Mossel (2012), Mossel, Rácz (2015))**.** *Every nearly non-manipulable strong voting game is nearly dictatorial.*

2.1.10. *Indeterminacy and chaos.* Condorcet's paradox asserts that the majority rule may lead to cyclic outcomes for three candidates. A stronger result was proved by McGarvey (1953): every asymmetric preference relation on m alternatives is the outcome of majority votes between pairs of alternatives for some individual rational preferences (namely, acyclic preferences) for a large number of voters. This property is referred to as *indeterminacy*. A stronger property is that when the individual order relations are chosen at random, the probability for every asymetric relation is bounded away from zero. This is called *stochastic indeterminacy*. Finally complete chaos refers to a situation that in the limit all the probabilities for asymmetric preference relations are the same $2^{-\binom{m}{2}}$.

Theorem 2.13 (Kalai (2004, 2007))**.** *(i) Social welfare functions based on voting games that aggregate information lead to complete indeterminacy. In particular this applies when the maximum Shapley value tends to zero.*

(ii) Social welfare functions based on voting games where the maximum Banzhaf value tends to zero leads to stochastic indeterminacy.

(iii) Social welfare function based on noise sensitive voting games leads to complete chaos.

2.1.11. *Discussion. Original contexts for some of the results.* Voting games are also called monotone Boolean functions and some of the results we discussed were proved in this context. Aggregation of information is also referred to as the sharp threshold phenomenon which is important in the study of random graphs, percolation theory and other areas. This was the main motivation for Junta's Junta theorem. Theorem 2.6 was studied in the context of distributed computing and the question of collective coin flipping: procedures allowing n agents to reach a random bit. Theorem 2.3 was studied in the context of the study of critical planar percolation. Friedgut's junta theorem was studied in the context of the combinatorics and probability of Boolean functions. The majority is stablest theorem was studied both in the context of hardness of approximation for the MAX CUT problem (see Section 2.5), and also in the context of social choice. Arrow's theorem and Theorem 2.11 had immense impact on theoretical economics and political science. There is a large body of literature with extensions and interpretations of Arrow's theorem and related phenomena were considered by many. Let me mention the more recent study of judgement aggregation, and also Peleg' books (1984, 2010) and Balisnki's books (1982, 2010) on voting methods that attempt to respond to the challenge posed by Arrow's theorem. Most proofs of the results discussed here go through Fourier analysis of Boolean functions that we discuss in Section 2.2.1.

Little more on cooperative games. I did not tell you about the most important solution concept in cooperative game theory, which is irrelevant to voting games, the core. The core of the games is an assignment of $v(N)$ to the n players so that the members of every coalition S get together at least $v(S)$. Bondareva and Shapley found necessary and sufficient conditions for the core to be non empty, closely related to linear programming duality. I also did not talked about games without side payments. There, $v(S)$ are sets of vectors which describe the possible payoffs for the player in S if they go together. A famous game with no side payment is Nash's bargaining problem for two players. Now, you are just one step away from one of the deepest and most beautiful results in game theory, Scarf's conditions (1967) for non-emptiness of the core.

But what about real-life elections? The relevance and interpretation of mathematical modeling and results regarding voting rules, games, economics and social science, is a fairly involved matter. It is interesting to examine some notions discussed here in the light of election polls which are often based on a more detailed model. Nate Silver's detailed forecasts provide a special opportunity. Silver computes the probability of victory for every candidate based on running many noisy simulations based on the outcomes of individual polls. (The way different polls are analyzed and how all the polls are aggregated together to give a model for voters' behavior is a separate interesting story.) The data in Silver's forecast contain an estimation for the event "recount" which essentially measures noise sensitivity and it would be interesting to compare noise sensitivity in this more realistic scenario to the simplistic model of i.i.d. voter's behavior. The data in Silver's forecast allow to estimate aggregation of information. Silver also computes certain power indices based on the probability for pivotality, again, under his model.

2.2. **Boolean functions and their Fourier analysis.**

We start with the discrete cube $\Omega_n = \{-1,1\}^n$. A Boolean function is a map $f : \Omega_n \to \{-1,1\}$.

Remark 2.14. A Boolean function represents a family of subsets of $[n] = \{1,2,\dots,n\}$ (also called hypergraph) which are central objects in extremal combinatorics. Of course, voting games are monotone Boolean functions. We also note that in agreement with Murphy's law, roughly half of the times it is convenient to consider additive notation, namely to regard $\{0,1\}^n$ as the discrete cube and Boolean functions as functions to $\{0,1\}$. (The translation is $0 \to 1$ and $1 \to -1$.)

2.2.1. *Fourier.* Every real function $f : \Omega_n \to \mathbb{R}$ can be expressed in terms of the Fourier–Walsh basis. We write here and for the rest of the paper $[n] = \{1,2,\dots,n\}$.

$$f = \sum \{\hat{f}(S) W_S : \ S \subset [n]\}, \tag{2.1}$$

where the *Fourier-Walsh function* W_S is simply the monomial $W_S(x_1,x_2,\dots,x_n) = \prod_{i \in S} x_i$.

Note that we have here 2^n functions, one for each subset S of $[n]$. The function W_S is simply the *parity* function for the variables in S. These functions form an orthonormal basis of $\mathbb{R}^{\Omega_n}$ with respect to the inner product

$$\langle f, g \rangle = \sum_{x \in \Omega_n} \mu(x) f(x) g(x).$$

The coefficients $\hat{f}(S) = \langle f, W_s \rangle$, $S \subset [n]$, in (2.1) are real numbers, called the *Fourier coefficients* of f.

Given a real function f on the discrete cube with Fourier expansion $f = \sum \{\hat{f}(S) W_S : \ S \subset [n]\}$, the noisy version of f, denoted by $T_\rho(f)$ is defined by $T_\rho(f) = \sum \{\hat{f}(S)(\rho)^{|S|} W_S : \ S \subset [n]\}$.

2.2.2. *Some more examples of Boolean functions. Boolean formulas, Boolean circuits, and projections.* (Here it is convenient to think about the additive convention.) Formulas and circuits allow to build complicated Boolean functions from simple ones and they have crucial importance in computational complexity. Starting with n variables $x_1, x_2, \dots, x_n$, a *literal* is a variable x_i or its negation $\neg x_i$. Every Boolean function can be written as a formula in conjunctive normal form, namely as

AND of ORs of literals. A *circuit* of depth d is defined inductively as follows. A circuit of depth zero is a literal. A circuit of depth one consists of an OR or AND *gate* applied to a set of literals, a circuit of depth k consists of an OR or AND gate applied to the outputs of circuits of depth $k-1$. (We can assume that gates in the odd levels are all OR gates and that the gates of the even levels are all AND gates.) The size of a circuit is the number of gates. Formulas are circuits where we allow to use the output of a gate as the input of only one other gates. Given a Boolean function $f(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$ we can look at its projection (also called trace) $g(x_1, x_2, \dots, x_n)$ on the first n variables. $g(x_1, x_2, \dots, x_n) = 1$ if there are values $a_1, a_2, \dots, a_m)$ (depending on the x_is) such that $f(x_1, x_2, \dots, x_n, a_1, a_2, \dots, a_m) = 1$. Monotone formulas and circuits are those where all linerals are variables. (No negation.)

Graph properties. A large important family of examples is obtained as follows. Consider a property P of graphs on m vertices. Let $n = m(m-1)/2$, associate Boolean variables with the n edges of the complete graph K_m, and represent every subgraph of K_m by a vector in Ω_n. The property P is now represented by a Boolean function on Ω_n. Even more generally, we can start with an arbitrary graph H with n edges and for every property P of subgraphs of H obtain a Boolean function of n variables based on P.

2.3. **Noise sensitivity everywhere (but mainly percolation).**

One thing we learned through the years is that noise sensitivity is (probably) a fairly common phenomenon. This is already indicated by Theorem 2.3. Proving noise sensitivity can be difficult. I will talk in this section about results on the critical planar percolation model, and conclude with a problem by Benjamini and Brieussel. I will not be able to review here many other noise-sensitivity results that justify the name of the section.

2.3.1. *Critical planar percolation.*

Theorem 2.15 (Benjamini, Kalai, and Schramm). *(1999) The crossing event for percolation is sensitive to $o(\log n)$ noise.*

Theorem 2.16 (Schramm and Steif (2011)). *The crossing event for percolation is sensitive to $o(n^c)$ noise, for some $c > 0$.*

Theorem 2.17 (Garban, Pete and Schramm (2010, 2013); Amazing!). *The crossing event for (hex) percolation is sensitive to $(n^{(3/4)-o(1)})$ noise. The spectral distribution has a scaling limit and it is supported by Cantor-like sets of Hausdorff dimension 3/4.*

Remark 2.18 (Connection to algorithms). The proof of Schramm and Steif is closely related to the model of computation of random decision trees. Decision tree complexity refers to a situation where given a Boolean function we would like to find its value by asking as few as possible questions about specific instances. Random decision trees allow to add randomization in the choice of the next question. These relations are explored in O'Donnell, Saks, Schramm, and Servatio (2005) and have been very useful in recent works in percolation theory.

Remark 2.19 (Connection to games). Critical planar percolation is closely related to the famous game of Hex. Peres, Schramm, Sheffield, and Wilson (2007) studied random turn-Hex where a coin-flip determines the identity of the next player to play. They found a simple but surprising observation that the value of the game when both players play the random-turn game optimally is the same as when both players play randomly. (This applies in much greater generality.) Richman considered such games which are auction-based turn. Namely, the players bid on who will play the next round. A surprising, very general analysis (Lazarus, Loeb, Propp, Ullman (1996)) shows that the value of the random-turn game is closely related to that of the auction-based game! Nash famously showed that for ordinary Hex, the first player wins but his proof gives no clue as to the winning strategy.

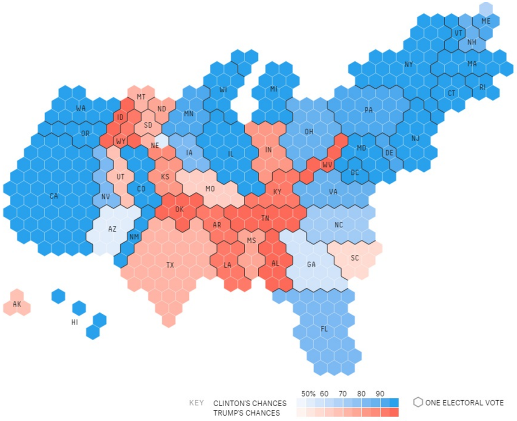

FIGURE 2. Hex based demonstration in Nate Silver's site

2.3.2. *Spectral distribution and Pivotal distribution.* Let f be a monotone Boolean function with n variables. We can associate to f two important probability distributions on subsets of $\{1, 2, \dots, n\}$.

The spectral distribution of f, $\mathcal{S}(f)$ gives a set S a probability $\hat{f}^2(S)$. Given $x \in \Omega_n$ the ith variable is *pivotal* if when we flip the value of x_i the value of f is flipped as well. (This is equivalent to the definition of pivotality we had for voting games.) The pivotality distribution $\mathcal{P}(f)$ gives a set S the probability that S is the set of pivotal variables. It is known that the first two moments of $\mathcal{S}$ and $\mathcal{P}$ agree.

Problem 1. (i) Find connections between $\mathcal{S}(f)$ and $\mathcal{P}(f)$ for all Boolean functions and for specific classes of Boolean functions.

Conjecture 2. Let f represents the crossing event in planar percolation. Show that $H(\mathcal{S})(f)) = O(I(f))$ and $H(\mathcal{P}(f)) = O(I(f))$. Here H is the entropy function.

Here H is the entropy function. The first inequality is a special case of the entropy-influence conjecture of Friedgut and Kalai (1996) that applies to general Boolean functions. The second inequality is not so general (it fails for the majority). The majority function f has an unusual property (that I call "the anomaly of majority") that $I(f) = \mathbb{E}((\mathcal{S}) = c\sqrt{n}$ is large, while most of the spectral weight of f is small, We note that if f is in **P** the pivotal distribution can be efficiently sampled. The spectral distribution can be efficiently sampled on a quantum computer (Section **??**).

2.3.3. *First passage percolation.* Consider an infinite planar grid where every edge is assigned a length: 1 with probability 1/2 and 2 with probability 1/2 (independently). This model of a random metric on the planar grid is called first-passage percolation. An old question is to understand what is the variance $V(n)$ of the distance D from $(0, 0)$ to $(n, 0)$? Now, let M be the median value of D and consider the Boolean function f describing the event "$D \geq M$". Is f noise sensitive?

Benjamini, Kalai and Schramm (2003) showed, more or less, that f is sensitive to logarithmic level of noise, and concluded that $V(n) = O(n/\log n)$. (The argument uses hypercontractivity and is similar to the argument for critical planar percolation.) To show that f is sensitive to noise level of n^δ for $\delta > 0$ would imply that $V(n) = O(n^{1-c})$. A very interesting question is whether methods used for critical planar percolation for obtaining stronger noise sensitivity results can also be applied here.

2.3.4. *A beautiful problem by Benjamini and Brieussel.* Consider n steps simple random walk (SRW) X_n on a Cayley graph of a finitely generated infinite group Γ. Refresh independently each step with probability ϵ, to get Y_n from X_n. Are there groups for which the positions at time n, X_n and Y_n are asymptotically independent? That is, the l_1 (total variation) distance between the chain (X_n, Y_n) and two independent copies (X'_n, X''_n) is going to 0, with n.

Note that on the line $\mathbb{Z}$, they are uniformally correlated, and therefore also on any group with a non trivial homomorphism to $\mathbb{R}$, or any group that has a finite index subgroups with a non trivial homomorphism to $\mathbb{R}$. On the free group and for any non-Liouville group, X_n and Y_n are correlated as well, but for a different reason: Both X_n and Y_n have nontrivial correlation with X_1. Itai Benjamini and Jeremie Brieussel conjecture that these are the only ways not to be noise sensitive. That is, if a Cayley graph is Liouville and the group does not have a finite index subgroup with a homomorphism to the reals, then the Cayley graph is noise sensitive for the simple random walk. In particular, the Grigorchuk group is noise sensitive for the simple random walk!

2.4. **Boolean complexity, Fourier and noise.**

2.4.1. **P** $\neq$ **NP** – *circuit version.* . The **P** $\neq$ **NP**-conjecture (in a slightly stronger form) asserts that the Boolean function described by the graph property of containing a Hamiltonian cycle, cannot be described by a polynomial-size circuit. Equivalently, the circuit form of the **NP** $\neq$ **P**-conjecture asserts that there are Boolean functions that can be described by polynomial size nondeterministic circuits, namely as the projection to n variables of a polynomial-size circuit, but cannot be described by polynomial size circuits. A Boolean function f is in **co-NP** if $-f$ is in **NP**.

Remark 2.20. Projection to n variables of a Boolean function in **co-NP** is believed to enlarge the family of functions even further. The resulting class is denoted by Π^2_P and the class of functions $-f$ when $f \in \Pi^2_P$ is denoted by Σ^2_P. By repeating the process of negating and projecting we reach a whole hierarchy of complexity classes, **PH**, called the polynomial hierarchy.

2.4.2. *Well below* **P**. The class **NC** describes Boolean functions that can be expressed by polynomial size polylogarithmically depth Boolean circuits. This class (among others) is used to model the notion of parallel computing. Considerably below, the class **AC**^{0} describes Boolean functions that can be expressed by bounded-depth polynomial-size circuit, where we allow AND ad OR gates to apply on more than two inputs. A celebrated result in computational complexity asserts that majority and parity do not belong to **AC**^{0} However, the noise-stability of majority implies that majority can be well approximated by functions in **AC**^{0}. We note that functions in **AC**^{0} are already very complex mathematical objects.

A monotone threshold circuit is a circuit built from gates which are are weighted majority functions (without negations). A general threshold circuit is a circuit built from gates which are threshold linear functions, i.e. we allow negative weights. **TC**^{0} (**MTC**^{0}) is the class of functions described by bounded depth polynomial size (monotone) threshold circuits.

2.4.3. *Some conjectures on noise sensitivity and bounded depth monotone threshold circuits.*

Conjecture 3 (Benjamini, Kalai, and Schramm (1999))**.** (i) Let f be a Boolean function described by monotone threshold circuit of size M and depth D. Then f is stable to $(1/t)$-noise where $t = (\log M)^{100D}$.

(ii) Let f be a monotone Boolean function described by a threshold circuit of size M and depth D. Then f is stable to $(1/t)$-noise where $t = (\log M)^{100D}$.

The constant 100 in the exponent is, of course, negotiable. In fact, replacing $100D$ with any function of D will be sufficient for most applications. The best we can hope for is that the conjectures are true if t behaves like $t = (\log M)^{D-1}$. Part (i) is plausible but looks very difficult. Part (ii) is quite reckless and may well be false. (See, however, Problem 5, below.) Note that the two parts differ "only" in the location of the word "monotone."

There are many Boolean functions that are very noise sensitive. A simple example is the recursive majority on threes, denoted by RM3 and defined as follows: Suppose that $n = 3^m$. Divide the variables into three equal parts. Compute the RM3 separately for each of these parts and apply majority to the three outcomes.

Conjecture 3 would have the following corollaries (C1)–(C4). Part (i) implies: (C1)– RM3 is not in **MTC**^{0}, and even C2 – RM3 cannot be approximated by a function in Monotone **MTC**^{0}. (A variant of (C1) is known by results of Yao (1989) and Goldmann and Hastad (1991), and these results motivated our conjecture.) (C2) already seems well beyond reach.

Part (ii) implies: (C3)– RM3 is not in **TC**^{0} and (C4)– RM3 cannot be approximated by a function in **TC**^{0}. Of course, we can replace RM3 with other noise-sensitive properties like the crossing event in planar percolation.

2.4.4. *Bounded depth Boolean circuit and the reverse Hastad conjecture.* Here is briefly the situation for **AC**^{0}.

Theorem 2.21. *(i) (Boppana (1984)): If f is a monotone Boolean function that can be described by a depth D size M monotone Boolean circuit then $I(f) \leq C(\log M)^{D-1}$.*

Here $I(f)$ denotes the total influence of f. Hastad switching lemma implies

(ii) (Hastad (1989) and Boppana (1997)) If f is a function that can be described by a depth D size M monotone Boolean circuit then $I(f) \leq C(\log M)^{D-1}$.

(iii) (Linial Mansour Nisan (1993); improved by Hastad (2001)): If f is a function that can be described by a depth D size M monotone Boolean circuit then $\{\sum \hat{f}^2(S) : |S| = t\}$ decays exponentially with t when $t > C(\log M)^{D-1}$.

We conjecture that functions with low influence can be approximated by low-depth small size circuits. A function g δ-approximates a function f if $|\mathbb{E}(f-g)^2| \leq \epsilon$.

Conjecture 4 (Benjamini, Kalai, and Schramm (1999))**.** For some absolute constant C the following holds. For every $\epsilon > 0$, a balanced Boolean function f can be ϵ-approximated by a circuit of depth d of size M where $(\log M)^{Cd} \leq I(f)$.

2.4.5. *Positive vs. Monotone.* We stated plausible while hard conjectures on functions in **MTC**^{0} and a reckless perhaps wrong conjectures on monotone functions in **MTC**^{0}. But we cannot present a single example of a monotone function in **TC**^{0} that is not in **MTC**^{0}. To separate the conjectures we need monotone functions in **TC**^{0} that cannot even be approximated in **MTC**^{0}. Ajtai and Gurevich (1987) proved that there are monotone functions in **AC**^{0} that are not in monotone **AC**^{0}.

Problem 5. (i) Are there monotone functions in **AC**^{0} that cannot be approximated by functions in Monotone **AC**^{0} ?

(ii) Are there monotone functions in **TC**^{0} that are not in **MTC**^{0}?

(iii) Are there monotone functions in **TC**^{0} that cannot be approximated by functions in **MTC**^{0}?

2.5. A small taste of PCP, hardness of approximation, and maxcut.

2.5.1. *Vertex cover and max cut.* A vertex cover of a graph G is a set of vertices such that every edge contains a vertex in the set. VERTEX COVER is the algorithmic problem of finding such a set of vertices of minimum size. Famously this problem is an **NP**-complete problem, in fact, it is one of the problems in Karp's original list. A matching in a graph is a set of edges such that every vertex is included in at most one edge. Given a graph G there is an easy efficient algorithm to find a maximal matching. Finding a maximal matching with r edges with respect to inclusion, gives us at the same time a vertex cover of size $2r$ and a guarantee that the minimum size of a vertex cover is at least r. A very natural question is to find an efficient algorithm for a better approximation. There is by now good evidence that this might not be possible. It is known to derive (Khot and Regev(2003)) from Khot's unique game conjecture (Khot (2002)).

A cut in a graph is a partition of the edges into two sets. The MAX CUT problem is the problem of finding a a cut with the maximum number of edges between the parts. Also this problem is **NP**-complete, and in Karp's list. The famous Goemans-Williamson algorithm based on semidefinite programming achieves α-approximation for max cut where $\alpha_{GM} = .878567$. A very natural question is to find an efficient algorithm for a better approximation. There is by now good evidence that this might not be possible.

2.5.2. *Unique games, the unique game conjecture, and the PCP theorem.* We have a connected graph and we want to color it with n colors. For every edge e we are given an orientation of the edge and a permutation π_e on the set of colors. In a good coloring of the edge if the tail is colored c then the head must be colored $\pi_e(c)$. It is easy to check efficiently if a global good coloring exists since coloring one vertex forces the coloring of all others.

Given ϵ, δ, n the unique game problem is to algorithmically decide between two scenarios (when we are promised that one of them holds.) Given a graph, G, a color set of size k and a permutation constraint for each edge.

(i) There is no coloring with more than ϵ fraction of the edges are colored good.

(ii) There is a coloring for which at least fraction $1 - \delta$ of the edges are colored good.

The unique game conjecture asserts that for every $\epsilon > 0$ and $\delta > 0$ it is NP-hard to decide between these two scenarios.

If one does not insist on the constraints being permutations and instead allows them to be of general form, then the above holds, and is called the PCP Theorem – one of the most celebrated theorems in the theory of computation.

Remark 2.22. A useful way to describe the situation (which also reflects the historical path leading to it) is in terms of a three-players games - there are two “provers” and a verifier. A verifier is trying to decide in which of the two cases he is in, and can communicate with two all powerful (non-communicating) provers. To do that, the verifier samples an edge, and sends one endpoint to each prover. Upon receiving their answers, the verifier checks that the two colors satisfy the constraint. The provers need to convince the verifier that a coloring exists by giving consistent answers to simultanous questions drawn at random.

2.5.3. *The theorem of Khot, Kindler, Mossel, and O'Donell.*

Theorem 2.23. *[Khot, Kindler, Mossel, O'Donnell (2007)] Let $\beta > \alpha_{GM}$ be a constant. Then an efficient β-approximation algorithm for* MAX CUT *implies an efficient algorithm for unique-games.*

The reduction relies on the majority is stablest theorem (Theorem 2.5) which was posed by Khot, Kindler, Mossel, and O'Donnell as a conjecture and later proved by Mossel, O'Donnell, and Oleszkiewicz (Theorem 2.5). This result belongs to the theory of hardness of approximation and probabilistically checkable proofs which is among the most important areas developed in computational complexity in the last three decades. For quite a few problems in Karp's original list of NP-complete problems (and many other problems added to the list) there is good evidence that the best efficient approximation is achieved by a known relatively simple algorithm. For a large class of problems it is even known (Raghavendra (2008)) (based on hardness of the unique game problem) that the best algorithm is either a very simple combinatorial algorithm (like for VERTEX COVER), or a more sophisticated application of semidefinite programming (like for MAX CUT). I will give a quick and very fragmanted taste on three ingredients of the proof of Theorem 2.23.

The noisy graph of the cube The proof of the hardness of max cut relative to unique games is based on the weighted graph whose vertices are the vertices of the descrete cube, all pairs are edges, and the weight of an edge between two vertices of distance k is $(1-p)^k p^{n-k}$. It turns out that in order to analyze the reduction, it suffices to study the structure of good cuts in this very special graph.

The least efficient error correcting codes. Error correcting codes have, for many decades, been among the most celebrated applications of mathematics with huge impact on technology. They also have a prominent role in theoretical computer science and in PCP theory. The particular code needed for max cut is the following: Encode a number k between 1 to n (and thus $\log n$) bits by a Boolean function - a dictatorship where the kth variable is the dictator!

Testing dictatorship An important ingredient of a PCP proof is "property testing", testing by looking at a bounded number of values if a Boolean function satisfies a certain property, or is very far from satisfying it. In our case we would like to test (with high probability of success) if a Boolean function is very far from dictatorship, or has substantial correlation with it. The test is the following: Choose x at random let $y = N_\epsilon(-x)$. Test if $f(x) = -f(y)$. For the majority function the probability that majority passes the test is roughly $\arccos(\epsilon - 1)$, majority is stablest theorem implies that anything that is more stable has large correlation with a dictator.

2.5.4. *Discussion: integrality gap and polytope integrality gap.* Given a graph G and nonnegative weights on its vertices set of vertices such that every edge contains a vertex in the set. The weighted version of vertex cover is the algorithmic problem of finding a set of vertices of minimum weight that covers all edges.

Minimize $w_1x_1 + w_2x_2 + \cdots + w_nx_n$ where $x = (x_1, x_2, \ldots, x_n)$ is a 0-1 vectors, subject to $x_i + x_j \geq 1$ for every edge $\{i,j\}$.

Of course, this more general problem is also NP-complete. The linear programming relaxation allows x_is to be real belonging to the interval [0,1]. The *integrality gap* for general vertex cover problems is 2 and given the solution to the linear programming problem you can just consider the set of vertices i so that $x_i \geq 1/2$. This will be a cover and the ratio between this cover and optimal one is at most 2. The integrality gap for the standard relaxation of max cut is $\log n$. Integrality gap is an important part of the picture in PCP theory. I conclude with a beautiful problem that I learned from Anna Karlin.

Consider the integrality gap (called the *polytope integrality gap*) between the covering problem and the linear programming relaxation when the graph G is fixed. In greater generality, consider a general covering problem of maximizing $c^t x$ subject to $Ax \leq b$ where A is integral matrix of nonnegative integers. Next, considered the integrality gap between 0-1 solutions and real solutions in $[0,1]$ when A and b are fixed (thus the feasible polyhedron is fixed, hence the name "polytope integrality gap") and only c (the objective function) varies.

The problem is if for vertex cover for every graph G and every vector of weights, there is an efficient algorithm achieving the polytope integrality gap. The same question can be asked for polytope integrality gap of arbitrary covering problems.

E-mail address: KALAI@MATH.HUJI.AC.IL