Pseudorandomness Program Notes & Open Problems

Spring 2017

Simons Institute

Feb 28, 2017

Contents

1 Expanders and Extractors Workshop (Jan 30–Feb 3)

Open Problems presented at the Simons Workshop on Expanders and Extractors.

Compiled by Noga Alon.

1.1 Partial Steiner systems of large girth, Nati Linial

Let us first recall that the *girth* of a graph G is the least integer g such that there is a set of g vertices in G that spans at least g edges. We seek an analogous notion for 3-uniform hypergraphs. In fact we only deal with *linear* hypergraphs H where every two hyperedges share at most one vertex. As defined by Erdős, the girth of H is the smallest integer $g \geq 4$ such that there is a set of g vertices in H that spans at least $g-2$ hyperedges. He conjectured that there exist Steiner Triple Systems with arbitrarily high girth, but despite considerable attempts over the years, the state of our knowledge concerning this problem is quite bad. I therefore formulate a variant that may be more accessible:

Question: Does there exist $c > 0$ and n-vertex 3-uniform hypergraphs with at least cn^2 hyperedges and arbitrarily high girth?

1.2 Cliques in near Ramanujan Graphs, Noga Alon

An (n, d, λ)-graph is a d-regular graph on n vertices so that all eigenvalues but the top one are in absolute value at most λ. Let G be an $(n, d, 100\sqrt{d})$-graph. It is known that:

1. There is a constant $c > 0$ so that if $d \geq cn^{2/3}$ then G contains a triangle.
2. The statement in (1) is tight up to the constant c.
3. There is a constant $c > 0$ so that if $d \geq cn^{4/5}$ then G contains a copy of K_4.

Open: Is (3) tight up to the value of c? That is, is there an $(n, d, 100\sqrt{d})$-graph containing no K_4, where $d = \Omega(n^{4/5})$? It is not even known whether or not there is such a graph with $n^{2/3} = o(d)$.

1.3 On the extractable entropy from zero-fixing sources, Gil Cohen

Let $n \geq k \geq 0$ be integers. An n-bit random variable X is called a *k-zero-fixing source* if there exists a subset of indices $R \subseteq [n]$ such that the marginal of X when projected to R is uniformly distributed, and the remaining bits of X are fixed to zero. The parameter k is called the entropy of X. Let $m = m(n, k)$ be the largest integer for which there exists a function $\mathsf{Ext}_k : \{0,1\}^n \to \{0,1\}^m$ with the following property: for any k-zero-fixing source X, $\mathsf{Ext}_k(X)$ is within statistical distance $1/5$ from uniform (here $1/5$ is an arbitrary choice of a small constant.) We stress that Ext_k does not get the description of the source X (namely, the set R) but rather a single sample from X. Such a function Ext_k is called an *extractor for k-zero-fixing sources*. We would like to have a better understanding of the function $m(n, k)$ that, informally speaking, captures the amount of extractable entropy from zero-fixing sources. Clearly, $m(n, k) \leq k$. By a straightforward counting argument, for any $k > \log_2 \log n + \log_2 \log \log n + \Omega(1)$, $m(n, k) = k - O(1)$. That is, when the entropy k is high enough, one can extract essentially all the entropy from the source. What about smaller entropies? For any k, $m(n, k) \geq 0.5 \log_2 k - O(1)$ as can be seen by considering the function that takes the Hamming weight of the input X modulo $\Theta(\sqrt{k})$ *[KZ06][RV13]*. That is to say, a logarithmic amount of entropy is always extractable. Interestingly, this simple function is optimal for small enough k. Put formally, $m(n, (\log^* n)^{2/3}) \leq 0.5 \log_2 k + O(1)$ *[CS15]*. The natural open problem is to understand the behavior of $m(n, k)$ and close the gap between $k = \Omega(\log \log n)$ and $k = O((\log^* n)^{2/3})$. In particular, it is not clear if $m(n, k)$ has a threshold behavior, namely, if there exists a function $\tau(n)$ such that for $k = \omega(\tau(n))$, $m(n, k) = k - O(1)$ whereas for $k = o(\tau(n))$, $m(n, k) \leq 0.5 \log_2 k + O(1)$.

1.4 Beating the expander mixing lemma for small sets, David Zuckerman

The expander mixing lemma asserts that in a d-regular graph G on n nodes, for sets S and T of size k, we have

$$|e(S,T) - dk^2/n| < \lambda k,$$

where $\lambda = \max(\lambda_2, -\lambda_n)$ is the largest nontrivial eigenvalue in absolute value. Even for optimal $\lambda = O(\sqrt{d})$, this is useless when $k < c\sqrt{n}$. On the other hand, when $d > (n/k)\log(n/k)$, most d-regular graphs achieve an upper bound of $O(k\sqrt{(dk/n)\log(n/k)})$. This is better than the expander mixing lemma for $k = o(n)$.

The problem is to give an upper bound $f(G,k)$ that is better than λk for some interesting graphs or for most or even many graphs. For dense graphs, this corresponds to two-source extractors. Bounds on line-point incidence graphs are also known. It would be extremely interesting to have a general method.

1.5 Spectral radius problem for free groups, Emmanuel Breuillard

If μ is a probability measure on a non-abelian free group F, let $\sigma(\mu) := \|T_\mu\|$ be the norm of the convolution operator on $\ell^2(F)$

$$T_\mu : f \mapsto \mu * f,$$

where $\mu * f(x) = \sum_{g\in G} f(g^{-1}x)\mu(x)$.

Is it true that for every $\epsilon > 0$ there is $\delta > 0$ such that for all probability measures μ on F, the condition $\sigma(\mu) > \epsilon$ implies that there is a coset xH of a cyclic subgroup H of F such that $\mu(xH) > \delta$?

1.6 Explicit Coding Power Series, Anup Rao

We are interested in giving an explicit description of a formal power-series over a finite field F with some nice properties. The motivation comes from several applications related to coding.

We say that a power series $P(X) = p_0 + p_1X + \ldots$ is ϵ-*sparse* if there is a finite k such that of the first k coefficients of $P(X)$, at most ϵk of them are non-zero.

Definition: $P(X)$ is an ϵ-*coding power series* if for every polynomial $g(X)$ with $0/1$ coefficients, the power series $(1 + Xg(X))P(X)$ is not ϵ-sparse.

Fact: For every $\epsilon > 0$, there is a finite field F for which a random power series $P(X)$ will be a coding power series with positive probability.

Open Problem: Give an explicit example of a coding power-series.

2 Cryptography using Weak Sources of Randomness (Feb 6–Feb 9)

Open Problems presented by Yevgeniy Dodis at the Simons Working Group on Cryptography using Weak Sources of Randomness.

Prepared by Siyao Guo.

2.1 Interactive proofs with imperfect randomness.

The class of languages which admit deterministic interactive proofs is $\mathbf{NP}$. The class of languages ($\mathbf{IP}$) which admit *probabilistic* interactive proofs is $\mathbf{PSPACE}$.

Question: Is $\mathbf{IP\text{-}weak} = \mathbf{IP}$? Can we simulate probabilistic interactive proofs using imperfect random sources?

Dodis et al. *[DOPS04]* (pdf) showed that block sources are sufficient to similate interactive proofs.

2.2 Extraction from limited "bit-coin source"

source parameterized by γ, n and b.

repeat the following steps until n b-bit blocks output:

1. sample random b-bit X
2. sample a coin which is 1 with probability $1 - \gamma$.
3. if coin=0, output X as next block and go to step 1.
4. if coin=1, ask attacker if he wants to block X or not
5. if block, don't output anything and go to step 1, else output X and go to step 1.

Goal: extract (for now 1) ε-unbiased bit from such $X_1 \dots X_n$

Known: impossible if A can block unbounded number of times.

So let's limit number of blocked times by t

Question 1: $b = 1$. given t, ε, γ, what is smallest n for which possible?

Question 2: given t, ε, γ, what is smallest alphabet b for which can set $n = t + 1$.

—

Consider the case of (information-theoretic) private-key encryption where parties wish to encrypt a b-bit value using shared secret key sampled from an imperfect random source X over n bits. Bosley and Dodis *[BD07]* (pdf) showed that if such scheme is secure, then one can deterministically extract $b - \log n$ bits from X. Hence, to a large extent, true randomness is inherent for encryption.

2.3 Separation between encryption and extraction.

They conjecture that extracting b bits from X is impossible.

For any extractor $\mathrm{Ext} : \{0,1\}^n \times \left(\{0,1\}^{\mathrm{poly}(n)}\right)^B \to \{0,1\}$, there exists distribution X over $\{0,1\}^n$ and $\mathrm{Enc} : \{0,1\}^n \times [B] \to \{0,1\}^{\mathrm{poly}(n)}$ such that

- for any $m_0 \neq m_1$, $\Delta(\mathrm{Enc}(X, m_0), \mathrm{Enc}(X, m_1)) = 0$,
- $\Delta(\mathrm{Ext}(X, \mathrm{Enc}(X, 1), \dots, \mathrm{Enc}(X, B)), U_1) > \Omega(1)$

where U_1 is the uniform distribution over $\{0,1\}$, $B = 2^b$ and Δ stands for statistical distance.

Bosley and Dodis *[BD07]* showed that above conjecture is true for $b \leq \log n - \log\log n$.

2.4 Is true randomness inherent for sharing schemes?

A randomize function $\mathrm{share}(m, X) \to (L, R)$ (which takes a message m over b bits as input and uses X as the random source) is a 2-out-of-2 secret sharing scheme if

- (Reconstruction) there exists an algorithm Rec such that

$$\text{for every } m,\ \Pr[\mathrm{Rec}(L, R) = m] = 1,$$

- (Privacy) for any $m' \neq m$, $\Delta(L(m), L(m')) = 0$ and $\Delta(R(m), R(m')) = 0$.

Question: If $\mathrm{share}(m, X)$ is a 2-out-of-2 secret sharing scheme, can we deterministically extract random bits from X?

More background in secret sharing can be found in the survey by Beimel *[Bei11]* (pdf).

2.5 Beating RT-bound using computational extractor.

Radhakrishnan and Ta-Shma *[RTaShma97]* (pdf) showed that any seeded extractor with error ε suffers from $2 \log 1/\varepsilon$ entropy loss (entropy loss is the amount of entropy in source and seed substracting output length). Motivated by bypassing this limitation, one approach is to consider computational extractor, whose output is only required to be computationally indistinguishable from uniformly random.

Dachman-Soled et al. *[DachmanSoledGKM12]* (pdf), together with the result of Dodis et al. *[DPW14]* (pdf) showed that any efficient computational extractor beating RT-bound implies one-way function.

Question: Can we construct an *efficient* computational extractor *beating RT-bound* based on one-way functions?

Krawczyk *[Kra10]* (pdf) used extract-then-expand approach and showed a computational extractor for medium-to-high entropy sources. More background and other approaches for constructing computational extractors can be found in Yevgeniy's slides and lecture note.

3 Learning Models of Mathematical Objects (Feb 21–23)

Notes from presentation by Russell Impagliazzo at the Simons Working Group on Learning Models of Mathematical Objects.

Prepared by Holden Lee.

3.1 Background

A theme that cuts across many domains of computer science and mathematics is to find simple representations of complex mathematical objects such as graphs, functions, or distributions on data. These representations need to capture how the object interacts with a class of tests, and to approximately determine the outcome of these tests.

For example, in machine learning, the object might be a distribution on data points, high dimensional real vectors, and the tests might be half-spaces. The goal would be to learn a simple representation of the data that determines the probability of any half-space or possibly intersections of half spaces. In computational complexity, the object might be a Boolean function or distribution on strings, and the tests are functions of low circuit complexity. In graph theory, the object is a large graph, and the tests are the cuts In the graph; the representation should determine approximately the size of any cut. In additive combinatorics, the object might be a function or distribution over an Abelian group, and the tests might be correlations with linear functions or polynomials.

The focus of the working group is to understand the common elements that underlie results in all of these areas, to use the connections between them to make existential results algorithmic, and to then use algorithmic versions of these results for new purposes. For example, can we use boosting, a technique from supervised learning, in an unsupervised context? Can we characterize the pseudo-entropy of distributions, a concept arising in cryptography? Do the properties of dense graphs "relativize" to sub-graphs of expanders?

In particular, we'll start from boosting, a technique in machine learning to go from weak learning to strong learning, i.e., taking an algorithm that learns a function only with a small correlation and making one that learns the function on almost all inputs. We'll show how boosting implies a general Hardcore Distribution Lemma, showing that any function that cannot be $1 - \delta$ approximated by simple functions has a sub-distribution of size δ where it has almost no correlation with simple functions. By starting from boosting, we will be able to show a constructive version of

this lemma. From the Hardcore Distribution lemma, we'll derive the Dense Model Theorem used by Green and Tao to show arbitrarily long arithmetic progressions in the primes. Again, by starting with boosting, we get a general algorithmic version of DMT. This algorithmic version can then be used to derive a general Weak Regularity Theorem, with that of Frieze and Kannan and analogs for sparse graphs as a special case.

Hopefully, at this point, the working group will segue from known connections to new connections, e.g., is there a strong boosting that implies strong regularity? Can algorithmic regularity lemmas be used in ML?

We won't assume any background and will develop everything from first principles using only simple calculations. Here's an optional reading list, and some papers we might refer to.

Papers with results we'll cover:

- Klivans and Servedio, Boosting and Hard-core Sets, FOCS 99.
- Omer Reingold, Luca Trevisan, Madhur Tulsiani, Salil P. Vadhan: Dense Subsets of Pseudorandom Sets. FOCS 2008: 76-85
- Luca Trevisan, Madhur Tulsiani, Salil P. Vadhan: Regularity, Boosting, and Efficiently Simulating Every High-Entropy Distribution. IEEE Conference on Computational Complexity 2009: 126-136
- Russell Impagliazzo, Algorithmic Dense Model Theorems and Weak Regularity
- Sita Gakkhar Russell Impagliazzo Valentine Kabanets. Hardcore Measures, Dense Models and Low Complexity Approximations
-

Bibliography:

We won't go through these papers explicitly, but they provide the context.

- Robert E. Schapire: The Strength of Weak Learnability (Extended Abstract). FOCS 1989: 28-33 : 01 June 2005 A desicion-theoretic generalization of on-line learning and an application to boosting Yoav Freund, Robert E. Schapire
- Yoav Freund, Robert E. Schapire: Game Theory, On-Line Prediction and Boosting. COLT 1996: 325-332
- Russell Impagliazzo: Hard-Core Distributions for Somewhat Hard Problems. FOCS 1995: 538-545
- Thomas Holenstein: Key agreement from weak bit agreement. STOC 2005: 664-673
- Boaz Barak, Ronen Shaltiel, Avi Wigderson: Computational Analogues of Entropy. RANDOM-APPROX 2003: 200-215
- Alan M. Frieze, Ravi Kannan: The Regularity Lemma and Approximation Schemes for Dense Problems. FOCS 1996: 12-20
- Noga Alon, Amin Coja-Oghlan, Hiêp Hàn, Mihyun Kang, Vojtech R"odl, Mathias Schacht: Quasi-Randomness and Algorithmic Regularity for Graphs with General Degree Distributions. SIAM J. Comput. 39(6): 2336-2362(2010)
- Noga Alon, Assaf Naor: Approximating the Cut-Norm via Grothendieck's Inequality. SIAM J. Comput. 35(4): 787-803 (2006)
- Green, Ben; Tao, Terence (2008). "The primes contain arbitrarily long arithmetic progressions". Annals of Mathematics. 167 (2): 481–547.
- Tao, Terence; Ziegler, Tamar (2008). "The primes contain arbitrarily long polynomial progressions". Acta Mathematica. 201 (2): 213–305

3.2 Big picture

We'll talk about several results which have different names in different fields. You probably know them, but don't know the same or related idea comes up in the other fields.

	Boosting	Hard-core lemma	Dense model theorem	Weak regularity	?
Area	ML	CC, Derandomization	Additive combinatorics, CC	Graph theory	
Credit	Shapiro, Freund-Schapire	Impagliazzo, Holenstein	Green-Tao, Barak-Shaltiel-Wigderson	Szemeredi, Frieze-Kannan	
Get	Circuit computing f $1-\delta$ of the time	"	Proof that set isn't δ-dense	"	
Unless	Weak learner fails on distribution of density $\Omega(\delta)$	Hard-core distribution	$\Omega(\delta)$-dense "model" indistinguishable from set	A model succinctly describing set	
Algorithm needed	Weak learner	"	Distinguisher	"	

We will take these theorems that we know to be true and show implications between them. Implications are due to...

1. Boosting $\Longrightarrow$ Hard-core: Klivans and Servedio.
2. Hard-core $\Longrightarrow$ Dense model: Impagliazzo
3. Dense model $\Longrightarrow$ Weak regularity: Trevisan-Tulsiani-Vadhan, Reingold-Trevisan-Tulsiani-Vadhan
4. Weak regularity $\Longrightarrow$ boosting: Trevisan-Tulsiani-Vadhan

What can we gain from looking at these connections?

1. Versatility: We can "retrofit" algorithms for one setting to get algorithms for the other settings.

   For example, there are many boosting algorithms. When you follow this progression, you get different quantitative and qualitative versions of dense model theorem and regularity.

2. Algorithmic and constructive results:

   There are nonconstructive versions using the min-max theorem for boosting, hard-core lemma, dense model theorem. We care about algorithmic versions.

   Note that the algorithmic result that we care about is different in the different settings. In ML we care about getting a function that computes a function much of the time. On the other side, we're really after the distribution where the weak learner fails, so that we get a model that succinctly describes the set.

   We pay attention to do the reductions in an algorithmic, not just an existential way.

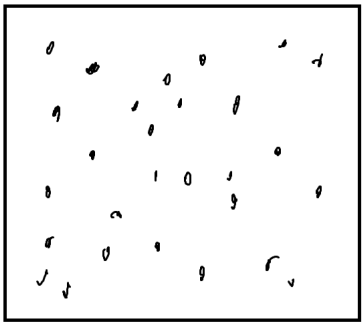
3. Using the dense model theorem for learning. Can we take a boosting technique and use it in an unsupervised way?

4. Generality: some things seem to be specific to a setting (density of graphs).

   But actually, weak regularity doesn't have anything to do with graphs being dense. We can relativize it to subgraphs of any graph. You can look at subgraphs of expanders, bipartite graphs, etc., and plug it in the same machinery. Likewise if you want to look at spectral norms rather than cuts.

Here is a cartoon:

1. Let X be a set, e.g. a distribution of points in the square. Let S be some distribution on points in X.

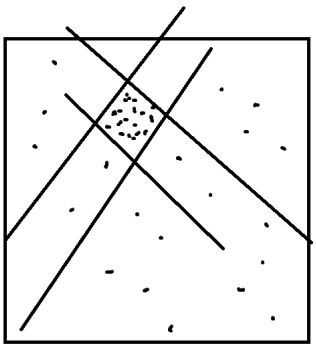


   Let $\mathcal{T}$ be a set of classifiers, ex. a set of half-planes.

   Let $\mathcal{F}_K\mathcal{T}$ be boolean functions on K functions in $\mathcal{T}$; here, partitions into polygonal regions by k half-planes.

   We want to pre-process the distribution to be able to answer queries in $\mathcal{F}_K\mathcal{T}$.

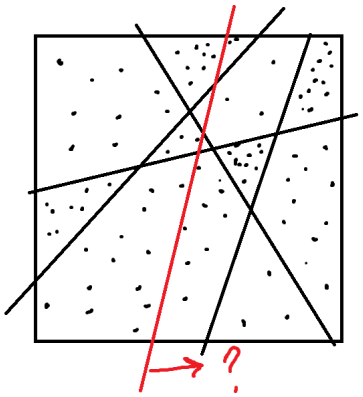
2. A violation of pseudo-density in this setting means there is a polygonal region with many more points from S than its volume, a "hot spot".



$$\text{Area(region)} < \delta \Pr_S(\text{region}) - \varepsilon.$$

3. A model is a partition into polygonal regions, with a probability distribution on regions. A simple model is defined by at most k lines.

   The property of a model is that we can estimate half-space probabilities ("given any half-space, what proportion of points of S are on one side of it?") by treating the points as if uniform within regions.



4. The algorithmic requirement in order to process the points to answer queries in $\mathcal{F}_K\mathcal{T}$ is: given a set of points sampled from S, and a set of points sampled from U, find a half-space that approximately maximizes the difference in probabilities for these two sets. The equivalent in boosting is a distinguishing algorithm.

Setting	Boosting	Hard-core measure	DMT/transference principle	Weak regularity
	WL: $\|\mu_i\| \geq 2\delta$, $\mu_i = g(h_1, \ldots, h_i, f)$, $h_{i+1} \in \mathcal{T}$, k iterations	Hardcore measure: $\mu_k = g(h_1, \ldots, h_k, f)$, $\|\mu_k\| \geq 2\delta$	Model: $\mu_k = g(h_1, \ldots, h_k, o)$, $\|\mu_k\| \geq \delta$	
	SL: $H = G(h_1, \ldots, h_k)$, $\Pr[H = f] \geq 1 - \delta$	Violation of hardness: $H = G(h_1, \ldots, h_k)$, $\Pr[H = f] > 1 - \delta$	Violation of pseudo-density $H = G(h_1, \ldots, h_k)$, $H(U) \leq \delta H(S) - \varepsilon$	
Assumption	WL never fails	Violation is impossible	Violation of pseudo-density is impossible	Actually dense
Conclusion	SL works	Hard-core measure exists, with same k, G, g	Model exists	Model exists
Algorithmic	Weak learner requirement	Approximately optimal weak learner	Approximately optimal distinguisher	

Setting	Boosting	Hard-core measure	DMT/transference principle	Weak regularity
	WL: $\|\mu_i\| \geq 2\delta$, $\mu_i = g(h_1, \ldots, h_i, f)$, $h_{i+1} \in \mathcal{T}$, k iterations	Hardcore measure: $\mu_k = g(h_1, \ldots, h_k, f)$, $\|\mu_k\| \geq 2\delta$	Model: $\mu_k = g(h_1, \ldots, h_k, o)$, $\|\mu_k\| \geq \delta$	
	SL: $H = G(h_1, \ldots, h_k)$, $\Pr[H = f] \geq 1 - \delta$	Violation of hardness: $H = G(h_1, \ldots, h_k)$, $\Pr[H = f] > 1 - \delta$	Violation of pseudo-density $H = G(h_1, \ldots, h_k)$, $H(U) \leq \delta H(S) - \varepsilon$	
Assumption	WL never fails	Violation is impossible	Violation of pseudo-density is impossible	Actually dense
Conclusion	SL works	Hard-core measure exists, with same k, G, g	Model exists	Model exists
Algorithmic	Weak learner requirement	Approximately optimal weak learner	Approximately optimal distinguisher	

Some comments:

1. Boosting: Note it's important that the δ here is the same; many boosting algorithms meet this criterion. The theorem says that "either weak learner fails or strong learner works."

   In boosting, we think of weak learner as never failing.

2. Hard-core measure lemma: The lemma says that either we can find hard-core measure, on which no function can compute the function f more than $1/2 + \delta$ of time; or find a function computing f $1 - \delta$ of the time.

   Here, we want to come up with the measure. Although the logical format is the same as boosting, here we assume that the violations never happen (there is no strong learner). Every boosting algorithm gives hard-core measure lemma with the same parameters, and with exactly the same way of "gluing" the functions. Sometime you care about computational complexity of G but not of g, or vice versa.

3. We can convert the hard-core measure theorem into the dense model theorem/transference principle (Tao and Ziegler).

   Here, we have a distribution we're trying to model. Either the distribution has pseudo-density property— there isn't a violation that's definable from k different properties from hypothesis class, where violation means that the expected value is much smaller on U than on S—or we get a model of density $\geq \delta$. Assuming that violation of pseudo-density does not happen, we get a model.

4. Weak regularity is just DMT except the distribution actually is dense. It's not so interesting that it has a dense model.

   What we get is that the dense model you get is simple, definable in terms of a small number of basic hypotheses.

   Sometimes we care about simplicity in the model, and sometimes simplicity in G.

5. Note the k is the same throughout. Reductions preserves k, and the functions h_i, G.

   We don't only have the fact that boosting implies hard-core lemma implies regularity lemma. We have the stronger result that whatever boosting algorithm you give me, I get a hard-core lemma and regularity lemma with the same parameters and algorithm. Thus we can pick the boosting algorithm that gives the best results for our application.

3.3 Setup

First we discuss the PAC learning model.

Let U be a set, and by abuse of notation, also a distribution on that set. (Think of U as the universe, the set of possible inputs.) For simplicity, take the distribution to be uniform. Let $f : U \to \{0, 1\}$ be a boolean function. A learning algorithm can request any number of points $(x, f(x))$ where $x \sim U$. The goal is to find a hypothesis h such that

$$\Pr_{x \sim U}[h(x) = f(x)] \geq 1 - \delta.$$

A for (U, f) with hypothesis class $\mathcal{H}$ is an algorithm such that given samples $(x, f(x)), x \sim U$, outputs $h \in \mathcal{H}$ (with high probability) such that

$$\Pr_{x \sim U}[h(x) = f(x)] \geq 1 - \delta.$$

(Typically, we say that the probability of success is $1 - \varepsilon$, ask for a strong learner for all $f \in \mathcal{F}$, and require it to run in time $\mathrm{poly}(1/\varepsilon, 1/\delta)$.)

In boosting, we assume that we have weak learners.

A ε- for (μ, f) with hypothesis class $\mathcal{H}$ is an algorithm such that given $(x, f(x)), x \sim \mu$, outputs h (with high probability) such that

$$\Pr_{x \sim \mu}[h(x) = f(x)] \geq \frac{1}{2} + \varepsilon.$$

It only has to output a function that is somewhat correlated with the right answer. Typically, we ask the weak learner to work on any distribution μ satisfying some assumptions.

In order to use a weak learner, we construct a routine that subsamples the distribution U to pass to pass to the weak learner.

Let $\mu : U \to [0, 1]$. Define the probability distribution[1]

$$D_\mu(x) = \frac{\mu(x)}{\sum_{x' \in U} \mu(x')}.$$

Think of this as rejection sampling: pick $x \sim U$, keep it with probability in $[0, 1]$, or else throw if back and repeat.

In order for this sampling to be efficient, we need μ to not be too small.

Define the of μ in U to be

$$|\mu| = \mathrm{E}_{x \in U}\, \mu(x).$$

We will use weak learners in the following context.

1. We will only run weak learners on distributions whose density is not too small (the dependence on δ is $|\mu| = \Omega(\delta)$). We don't want to run a weak learner on a distribution of very low density, because the time to simulate the distribution is inversely proportional to the density.
2. We ask the weak learners to output a function in a given class $h \in \mathcal{T}$.

   Then it will turn out that that both the measures that we run the weak learners on, and the final hypothesis, will be describable using $\mathcal{F}_l\mathcal{T}$ (see below), for some class $\mathcal{F}$.

Say that a set $\mathcal{T}$ of functions $U \to \{0, 1\}$ form a class if $f \in \mathcal{T}$ implies $1 - f \in \mathcal{T}$.

Let $\mathcal{F}$ be a class of boolean functions. Define the class of functions

$$\mathcal{F}_k\mathcal{T} = \{f(h_1(x), \ldots, h_k(x)) : f \in \mathcal{F}, h_1, \ldots, h_k \in \mathcal{T}\}.$$

[1] When U is not uniform and has distribution $u(x)$, this is $\frac{\mu(x)u(x)}{\sum_{x' \in U} \mu(x')u(x')}$.

3.4 Boosting and the Hard-core lemma

The first boosting algorithm we give is totally ridiculous from the ML point of view. For people who work on weak regularity on graphs this is the natural version, and leads to the standard versions of results.

We will take $\mathcal{F}$ to be the set of all boolean functions, so given hypotheses $h_1, \dots, h_k$, we can choose the best predictor using $h_1(x), \dots, h_k(x)$.

[Boosting with decision trees][thm:boosting] Let U be a distribution, $\mathcal{T}$ a class of boolean functions $U \to \{0,1\}$, $\mathcal{F}$ the class of all boolean functions. Let $f : U \to \{0,1\}$ be a given function (which we are trying to learn).

1. Suppose that there is a δ-weak learner such that given any distribution μ on U with $|\mu| \geq 2\delta$, it produces $h \in \mathcal{T}$ such that
$$\Pr_{x\sim\mu}[h(x) = f(x)] \geq \frac{1}{2} + \varepsilon.$$
2. Then there is a strong learner that produces $h \in \mathcal{F}_k\mathcal{T}$ with $k \leq \lceil 1/\varepsilon^2\delta^2 \rceil$ such that[2]
$$\Pr_{x\sim U}[h(x) = f(x)] \geq 1 - \delta.$$

[Hard-core lemma] [thm:hardcore] Let U be a distribution, $\mathcal{T}$ a class of boolean functions $U \to \{0,1\}$, $\mathcal{F}$ the class of all boolean functions.

Then either

1. There exists $h \in \mathcal{F}_k\mathcal{T}$ such that
$$\Pr_{x\sim U}[h(x) = f(x)] \geq 1 - \delta,$$
where $k \leq 1/\varepsilon^2\delta^2$, or
2. (There exists a hard-core distribution.) There exists $|\mu| \geq 2\delta$ on U, such that for all $h \in \mathcal{T}$,
$$\Pr_{x\sim\mu}[h(x) = f(x)] \leq \frac{1}{2} + \varepsilon.$$

Note it is important for us to keep track of the size of the hardcore distribution, which is $\geq 2\delta$ here. Different boosting algorithms will give the result for different classes of functions $\mathcal{F}$.

[Proof of hard-core lemma [thm:hardcore] from boosting [thm:boosting]] Let weak learner be exhaustive search over $\mathcal{T}$. The weak learner operates on distributions $|\mu_i| \geq 2\delta$. If it always produces h_i with bias $\geq \delta$, then continue and obtain the strong learner: we get some $H \in \mathcal{F}_k\mathcal{T}$ such that $H(x) = f(x)$ with probability $1 - \delta$.

If at some step i our exhaustive search algorithm gets stuck, we get a distribution μ_i that's hard-core.

3.5 Dense model theorem

For a set $S \subseteq U$ and a function $T : U \to \{0,1\}$, let $T(S) := \mathrm{E}_{x\in S}\, T(x)$. (For a measure $\mu : U \to [0,1]$, also write $T(\mu) = \mathrm{E}_{x\sim\mu}\, T(x)$.)

Let $S \subseteq U$ be a subset, and let $\mathcal{T}$ be a set of tests. S is if for all $T \in \mathcal{T}$,
$$T(U) \geq \delta T(S) - \varepsilon.$$

Think of saying that the tests $\mathcal{T}$ don't reveal that the set S is small.

[2] We ignore sample complexity here. In reality, because we only see U from samples, we need to think about generalization. If the VC-dimension of $\mathcal{T}$ is d, then the VC-dimension of $\mathcal{F}_k\mathcal{H}$ is at most k^d. In ML we don't want to take $\mathcal{F}$ to be the class of all boolean functions. For this theorem, let's just assume we are actually given all pairs $(x, f(x))$.

1. One way of being pseudo-dense is to actually be dense.
2. Another, one step removed, is that there's a set R (or more generally, a measure μ) that's indistibguishable from S by $\mathcal{T}$, and such that R occupies at least a δ fraction of U.
•

For two distributions μ_1, μ_2 on U, we say that μ_1, μ_2 are indistinguishable by tests in $\mathcal{T}$ up to ε, written $\mu_1 \sim_{\mathcal{T}} \mu_2$ within ε, if for every $T \in \mathcal{T}$,

$$|\operatorname{E}_{\mu_1} T - \operatorname{E}_{\mu_2} T| \leq \varepsilon.$$

[Dense model theorem][thm:dmt] Let $\mathcal{T}$ be a class of tests $U \to \{0,1\}$.

If S is (ε, δ)-pseudodense against $F_k\mathcal{T}$, $k = O(1/\varepsilon^2\delta^2)$ then there exists μ, $\mu \in F_k\mathcal{T}$ such that $|\mu| \geq \frac{\delta}{1+\delta} - O(\varepsilon)$ and $D_\mu \sim_{\mathcal{T}} S$ to within $O(\varepsilon/\delta)$.

The idea in the proof is to use the Hard-core lemma, with the hard function being membership in S.

Let U' be the following distribution: let $\delta' = \frac{\delta}{1+\delta}$ and

1. with probability δ', take $x \in S$ and output $(0, x)$
2. with probability $1 - \delta'$, take $x \in U$ and output $(1, x)$.

Define a test $T \in \mathcal{T}$ to operate on an example (y, x) by $T(y, x) = T(x)$. For $T \in \mathcal{F}_k\mathcal{T}$,

$$\begin{aligned}
\Pr_{U'}[T((y,x)) = y] &= \delta' T(S) + (1-\delta')(1 - T(U)) \\
&= 1 - \delta' + \delta'(T(S)) - (1-\delta')T(U) \\
&= 1 - \delta' + \frac{1}{1+\delta}(\delta T(S) - T(U)) \\
&\leq 1 - \delta' + \varepsilon.
\end{aligned}$$

No test in $\mathcal{F}_k\mathcal{T}$ can be correct with probability $> \delta' - \varepsilon$. By the Hard-core Lemma [thm:hardcore], there exists $|\mu'| \geq 2(\delta' - \varepsilon)$ such that for any $T \in \mathcal{T}$, $\Pr_{(x,y)\sim U'}[T(x) = y] \leq \frac{1}{2} + \varepsilon$.

In order for μ' to be hardcore, it must be split approximately evenly between U and S (up to ε); otherwise; we could have an advantage by predicting constant 0 or 1. Thus each part has at least $2(\delta' - \varepsilon)(1/2 - \varepsilon) = \delta'(1 - O(\varepsilon/\delta))$ of the mass. Then

$$D_{\mu'|_U} \sim_{O(\varepsilon)} D_{\mu'|_S} \sim_{O(\varepsilon/\delta)} S.$$

3.6 Proof for boosting

[Proof of Theorem [thm:boosting]] The algorithm is as follows. Let $WL(\mu)$ denote the weak learner operating on (μ, f).

Let μ_0 be constant 1, $i = 0$.

While $|\mu_i| \geq 2\delta$, do

- $h_{i+1} \leftarrow\!\shortmid WL(\mu_i)$.
- Partition U according to values of $h_1, \ldots, h_i$.

  Let $h_{1:i}(x) := (h_1(x), \ldots, h_i(x)) \in \{0,1\}^i$, and let $B_i(x)$ be the "block" that x is in,

  $$B_i(x) = h_{1:i}^{-1}(h_{1:i}(x)) = \{y \in U : h_{1:i}(x) = h_{1:i}(y)\}.$$

  For a set B, let $\operatorname{Maj}(B)$ denote the majority value of f on B.

- Define μ_{i+1} by

$$\mu_{i+1}(x) = \begin{cases} \frac{1-p_{\mathrm{Maj},B_i(x)}}{p_{\mathrm{Maj},B_i(x)}}, & \text{if } f(x) = \mathrm{Maj}(B_i(x)) \\ 1, & \text{otherwise} \end{cases}$$

  where $p_{\mathrm{Maj},B} = \Pr(f(y) = \mathrm{Maj}(B) | y \in B)$, the proportion of the majority in B.

- $i \leftarrow\!\shortmid i + 1$.

Finally, return $H_i(x) = \mathrm{Maj}(B_i(x))$, i.e., look at the block that x is in, and choose the majority value.

Note that the measure μ_{i+1} rebalances each block B_i such that conditioned on y being in a block $B_i(x)$,

$$\Pr_{y\sim\mu_{i+1}}(f(y) = 1 | y \in B_i(x)) = \Pr_{y\sim\mu_{i+1}}(f(y) = 0 | y \in B_i(x)) = \frac{1}{2}.$$

Indeed, we have

$$\begin{aligned} \mathrm{E}_{y\sim U}[\mathbf{1}_{f(y)=1}\mu_{i+1}(y) | y \in B_i(x)] &= p_{\mathrm{Maj},B_i(x)} \frac{1 - p_{\mathrm{Maj},B_i(x)}}{p_{\mathrm{Maj},B_i(x)}} = 1 - p_{\mathrm{Maj},B_i(x)} \\ \mathrm{E}_{y\sim U}[\mathbf{1}_{f(y)=0}\mu_{i+1}(y) | y \in B_i(x)] &= \left(1 - p_{\mathrm{Maj},B_i(x)}\right) \cdot 1 = 1 - p_{\mathrm{Maj},B_i(x)} \\ |\mu_{i+1}| = \mathrm{E}_{y\sim U}[\mu_{i+1}(y)] &= \sum_{\text{block } B_i} [2(1 - p_{\mathrm{Maj},B_i}) \Pr(B_i)] \\ &\geq 2(1 - p_{\mathrm{Maj},U}). \end{aligned}$$

Note that if $|\mu_{i+1}| \leq 2\delta$, then $\Pr_{x\in X}[H_i = f] \geq 1 - \delta$, and we are done. (We stop before we have to apply the weak learner to a distribution of density $< \delta$.)

We need to show this method terminates in a bounded number of steps.

Consider the potential function

$$\varphi_i = \mathrm{E}_{x\sim U}[(\Pr[f = 1 | B_i(x)])^2] = \mathrm{E}_{x\sim U}[\mathrm{E}[f|B_i]^2]$$

(Think of B_i as a partition; for a partition, $\mathrm{E}[f|P]$ is a function of x that takes x to the average value in the atom of the partition that contains x.) Note this have value in $[0, 1]$ and is maximized if f is constant on every block. We show every iteration increases this potential function by at least a fixed amount, $(\varepsilon\delta)^2$. Fix a block B in the partition. Define $p, q, \alpha_+, \alpha_-, p_0, p_1$ as follows.

$$\begin{aligned} p &= \Pr[f = 1 | B] \\ q &= \Pr[h_{i+1} = 1 | B] \\ q + \alpha_+ &= \Pr[h_{i+1} = 1 | B, f = 1] \\ q - \alpha_- &= \Pr[h_{i+1} = 1 | B, f = 0] \\ \alpha_+ p &= \alpha_-(1 - p) \text{ by conservation} \\ p_0 = \Pr[f = 1 | h = 0, B] &= \frac{\Pr[f = 1 \wedge h = 0 | B]}{\Pr[h = 0 | B]} = \frac{p(1 - q - \alpha_+)}{1 - q} \\ p_1 = \Pr[f = 1 | h = 1, B] &= \frac{\Pr[f = 1 \wedge h = 1 | B]}{\Pr[h = 1 | B]} = \frac{p(q + \alpha_+)}{q} \\ \mathrm{E}_{x\in B}[\mathrm{E}[f|B_{i+1}]^2] &= qp_1^2 + (1 - q)p_0^2 = p^2 \left(\frac{(q + \alpha_+)^2}{q} + \frac{(1 - q - \alpha_+)^2}{1 - q} \right) \\ &= p^2 \left(\left(q + 2\alpha_+ + \frac{\alpha_+^2}{q} \right) + \left(1 - q - 2\alpha_+ + \frac{\alpha_+^2}{1 - q} \right) \right) \\ &= p^2 \left(1 + \frac{\alpha_+^2}{q} + \frac{\alpha_+^2}{1 - q} \right) \\ &\geq p^2 + 4p^2\alpha_+^2 \geq p^2 + \alpha_+^2 \\ \mathrm{E}[f|B_{i+1}]^2 - \mathrm{E}[f|B_i]^2 &= \alpha_+^2(B_i(x)). \end{aligned}$$

Assume WLOG that $\mathrm{Maj}(B_i(x)) = 1$. (Otherwise the LHS is smaller.)

$$\begin{aligned}
\mathrm{E}_{x\in B}[\mu(x)((-1)^{(h(x)\neq f(x))})] = &\quad p\left(\frac{1-p}{p}\right)[(q+\alpha_+)-(1-q-\alpha_+)] \quad (f=1)\\
&+(1-p)1[1-(1-\alpha_-)-(q-\alpha_-)] \qquad (f=0)\\
&=(1-p)(2\alpha_+ + 2\alpha_-)\\
&=2\alpha_+(1-p)+2\alpha_+ p = 2\alpha_+\\
\mathrm{E}_{x\sim U}\, 2\alpha_+(B_i(x)) &= \mathrm{E}_{x\sim U}[\mu(x)((-1)^{h(x)\neq f(x)})]\\
&\geq \varepsilon|\mu| \geq 2\delta\varepsilon\\
\varphi_{i+1}-\varphi_i &\geq \mathrm{E}_{x\sim U}[\mathrm{E}[f|B_{i+1}]^2 - \mathrm{E}[f|B_i]^2]\\
&\geq \mathrm{E}_{x\sim U}\,\alpha_+^2(B_i(x)) \geq (\delta\varepsilon)^2.
\end{aligned}$$

• Because φ_i is always in $[0,1]$, the number of iterations is at most $k \leq (\delta\varepsilon)^2$.

3.7 Comments, Regularity lemmas

Some comments:

1. All you get from this proof is a decision tree; the complexity is exponential in k. This is a bug, not a feature.

   In complexity terms, we don't get good hard-core measure, because the circuit size for the outer function G is 2^k. A better boosting algorithm would give G have smaller complexity. If your stopping point is the hard-core lemma, this is not the boosting algorithm you want. For the dense model theorem, this is fine because all you care about is size of k, not the complexity of G.

   There is another boosting algorithm which gives a weighted majority function, which is a simpler function. A weighted majority can be converted into a decision tree, but not vice versa.

2. This potential function matches this boosting algorithm. Other boosting algorithms can be analyzed with other potential functions. This is like the potential function used most in graph theory. Key property: you can't make negative progress; you always go forwards.

3. For Szemeredi regularity, we need a stronger boosting theorem. Suppose we get stuck at some step: no function correlates globally, but there are many blocks where we can find functions that correlate with the function inside that block. If in ε fraction of blocks we find functions that correlate, partition them based on all the values of these functions, and repeat.

   In one step we've gone from order of 2^k to order of 2^{2^k} buckets, and increased the potential function by a polynomial in terms of ε, δ. This is a familiar argument; we can only go $\frac{1}{\varepsilon}$ iterations before we terminate. This time, the number of sets is a tower depending on ε.

4. Regularity lemmas:

   Fix a set of vertices V of set n. Let U be edges in complete graph on V. (We can also consider the case when U is not the complete graph, ex. U is the edges in d-regular expander on V.)

   The underlying set we care about is the set of cuts defined by $A, B \subseteq V$ where $A \cap B = \phi$; there are 3^k of them.

   If $|E| \geq \delta\binom{n}{2}$, the generic regularity lemma says there exists $\mu = G(T_1, \ldots, T_k)$, where $k = O(1/\varepsilon^2\delta^2)$, that is a good predictor the number of edges of any cut in the graph. Use the T's to divide the vertices into 3^k subsets such that μ is a constant on every pair of subsets.

   $$\frac{E_G(A,B)}{|E_G|} \approx_\varepsilon \sum_{i,j} \mu_{ij} \frac{|A\cap A_i||B\cap B_j|}{|V|^2}.$$

   This is the weak regularity of Frieze-Kannan. For Szemeredi we need the stronger boosting lemma (see previous point).

We can also do something similar with G a subset of an expander. The expander mixing lemma gives an error term.

References

[Bei11] Amos Beimel. Secret-sharing schemes: A survey. In *Coding and Cryptology - Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings*, 11–46. 2011. URL: http://dx.doi.org/10.1007/978-3-642-20901-7_2, doi:10.1007/978-3-642-20901-7_2.

[BD07] Carl Bosley and Yevgeniy Dodis. Does privacy require true randomness? In *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, 1–20. 2007. URL: http://dx.doi.org/10.1007/978-3-540-70936-7_1, doi:10.1007/978-3-540-70936-7_1.

[CS15] Gil Cohen and Igor Shinkar. Zero-fixing extractors for sub-logarithmic entropy. In *International Colloquium on Automata, Languages, and Programming*, 343–354. Springer, 2015.

[DachmanSoledGKM12] Dana Dachman-Soled, Rosario Gennaro, Hugo Krawczyk, and Tal Malkin. Computational extractors and pseudorandomness. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, 383–403. 2012. URL: http://dx.doi.org/10.1007/978-3-642-28914-9_22, doi:10.1007/978-3-642-28914-9_22.

[DOPS04] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, 196–205. 2004. URL: http://dx.doi.org/10.1109/FOCS.2004.44, doi:10.1109/FOCS.2004.44.

[DPW14] Yevgeniy Dodis, Krzysztof Pietrzak, and Daniel Wichs. Key derivation without entropy waste. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, 93–110. 2014. URL: http://dx.doi.org/10.1007/978-3-642-55220-5_6, doi:10.1007/978-3-642-55220-5_6.

[KZ06] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2006.

[Kra10] Hugo Krawczyk. Cryptographic extraction and key derivation: the HKDF scheme. In *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, 631–648. 2010. URL: http://dx.doi.org/10.1007/978-3-642-14623-7_34, doi:10.1007/978-3-642-14623-7_34.

[RTaShma97] Jaikumar Radhakrishnan and Amnon Ta-Shma. Tight bounds for depth-two superconcentrators. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, 585–594. 1997. URL: http://dx.doi.org/10.1109/SFCS.1997.646148, doi:10.1109/SFCS.1997.646148.

[RV13] Yakir Reshef and Salil Vadhan. On extractors and exposure-resilient functions for sublogarithmic entropy. *Random Structures & Algorithms*, 42(3):386–401, 2013.