Contents

Wiki page: `https://simons-institute.github.io/pseudorandomness/groups/modelstructure.html`

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): 481547.

- Tao, Terence; Ziegler, Tamar (2008). "The primes contain arbitrarily long polynomial progressions". Acta Mathematica. 201 (2): 213305

1 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.

2 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

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

Theorem 2.1. *A **strong learner** 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*

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

(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 $\mathrm{poly}(\frac{1}{\varepsilon}, \frac{1}{\delta})$.)

In boosting, we assume that we have weak learners.

Theorem 2.2. *A ε-**weak learner** 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*

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

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.

Definition 2.3: Let $\mu : U \to [0, 1]$. Define the probability distribution

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

[1]

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.

Definition 2.4: Define the **density** of μ in U to be

$$|\mu| = \mathop{\mathbb{E}}_{x\in U} \mu(x).$$

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 $\mathcal{F}$.

Definition 2.5: 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 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 $h_1(x), \dots, h_k(x)$.

Theorem 3.1 (Boosting with decision trees)**.** *Let U be a distribution, $\mathcal{T}$ a class of boolean functions $U \to \{0,1\}$. 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*

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

2. *Then there is a strong learner that produces $h \in \mathcal{F}_k\mathcal{T}$ with $k \leq \left\lceil \frac{1}{\varepsilon^2\delta^2} \right\rceil$ such that*

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

[2]

Theorem 3.2 (Hard-core lemma)**.** *Let U be a distribution, $\mathcal{T}$ a class of boolean functions $U \to \{0,1\}$.*

Then either

1. *There exists $h \in \mathcal{F}_k\mathcal{T}$ such that*

$$\mathbb{P}_{x\sim U}[h(x) = f(x)] \geq 1 - \delta,$$

   where $k \leq \frac{1}{\varepsilon^2\delta^2}$, or

2. *(There exists a hard-core distribution.) There exists $|\mu| \geq 2\delta$ on U, such that for all $h \in \mathcal{T}$,*

$$\mathbb{P}_{x\sim\mu}[h(x) = f(x)] \leq \frac{1}{2} + \varepsilon.$$

Proof of hard-core lemma 3.2 from boosting 3.1. 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. □

[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 $(x, f(x))$.

Proof of Theorem 3.1. 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 X 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 $\mathsf{Maj}(B)$ denote the majority value of f on B.
- Define μ_{i+1} by

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

  where $p_{\mathsf{Maj},B} = \mathbb{P}(f(y) = \mathsf{Maj}(B) | y \in B)$, the proportion of the majority in B.
- $i \leftarrow\!\shortmid i + 1$.

Finally, return $H_i(x) = \mathsf{Maj}(B_i(x))$, 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 $B_i(x)$,

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

Indeed, we have

$$\mathop{\mathbb{E}}_{y\sim U}[\mathbb{1}_{f(y)=1}\mu_{i+1}(y) | y \in B_i(x)] = p_{\mathsf{Maj},B_i(x)} \frac{1 - p_{\mathsf{Maj},B_i(x)}}{p_{\mathsf{Maj},B_i(x)}} = 1 - p_{\mathsf{Maj},B_i(x)} \tag{1}$$

$$\mathop{\mathbb{E}}_{y\sim U}[\mathbb{1}_{f(y)=0}\mu_{i+1}(y) | y \in B_i(x)] = \left(1 - p_{\mathsf{Maj},B_i(x)}\right) \cdot 1 = 1 - p_{\mathsf{Maj},B_i(x)} \tag{2}$$

$$|\mu_{i+1}| = \mathop{\mathbb{E}}_{y\sim U}[\mu_{i+1}(y)] = \sum_{\text{block } B_i} [2(1 - p_{\mathsf{Maj},B_i})\mathbb{P}(B_i)] \tag{3}$$

$$\geq 2(1 - p_{\mathsf{Maj},U}). \tag{4}$$

Note that if $|\mu_{i+1}| \leq 2\delta$, then $\mathbb{P}_{x\in X}[H_i = f] \geq 1 - \delta$, 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

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

(Think of B_i as a partition; for a partition, $\mathbb{E}[f|P]$ 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.

$$p = \mathbb{P}[f = 1|B] \tag{5}$$
$$q = \mathbb{P}[h_{i+1} = 1|B] \tag{6}$$
$$q + \alpha_+ = \mathbb{P}[h_{i+1} = 1|B, f = 1] \tag{7}$$
$$q - \alpha_- = \mathbb{P}[h_{i+1} = 1|B, f = 0] \tag{8}$$
$$\alpha_+ p = \alpha_-(1-p) \text{ by conservation} \tag{9}$$
$$p_0 = \mathbb{P}[f = 1|h = 0, B] = \frac{\mathbb{P}[f = 1 \wedge h = 0|B]}{\mathbb{P}[h = 0|B]} = \frac{p(1-q-\alpha_+)}{1-q} \tag{10}$$
$$p_1 = \mathbb{P}[f = 1|h = 1, B] = \frac{\mathbb{P}[f = 1 \wedge h = 1|B]}{\mathbb{P}[h = 1|B]} = \frac{p(q+\alpha_+)}{q} \tag{11}$$
$$\mathbb{E}_{x\in B}[\mathbb{E}[f|B_{i+1}]^2] = qp_1^2 + (1-q)p_0^2 = p^2\left(\frac{(q+\alpha_+)^2}{q} + \frac{(1-q-\alpha_+)^2}{1-q}\right) \tag{12}$$
$$= p^2\left(\left(q + 2\alpha_+ + \frac{\alpha_+^2}{q}\right) + \left(1 - q - 2\alpha_+ + \frac{\alpha_+^2}{1-q}\right)\right) \tag{13}$$
$$= p^2\left(1 + \frac{\alpha_+^2}{q} + \frac{\alpha_+^2}{1-q}\right) \tag{14}$$
$$\geq p^2 + 4p^2\alpha_+^2 \geq p^2 + \alpha_+^2 \tag{15}$$
$$\mathbb{E}[f|B_{i+1}]^2 - \mathbb{E}[f|B_i]^2 = \alpha_+^2(B_i(x)). \tag{16}$$

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

$$\mathop{\mathbb{E}}_{x\in B}[\mu(x)((-1)^{(h(x)\neq f(x))})] = \quad p\left(\frac{1-p}{p}\right)[(q+\alpha_+) - (1-q-\alpha_+)] \qquad (f=1) \tag{17}$$
$$+ (1-p)1[1 - (1-\alpha_-) - (q-\alpha_-)] \qquad (f=0) \tag{18}$$
$$= (1-p)(2\alpha_+ + 2\alpha_-) \tag{19}$$
$$= 2\alpha_+(1-p) + 2\alpha_+ p = 2\alpha_+ \tag{20}$$
$$\mathop{\mathbb{E}}_{x\sim U} 2\alpha_+(B_i(x)) = \mathop{\mathbb{E}}_{x\sim U}[\mu(x)((-1)^{h(x)\neq f(x)})] \tag{21}$$
$$\geq \varepsilon|\mu| \geq 2\delta\varepsilon \tag{22}$$
$$\varphi_{i+1} - \varphi_i \geq \mathop{\mathbb{E}}_{x\sim U}[\mathbb{E}[f|B_{i+1}]^2 - \mathbb{E}[f|B_i]^2] \tag{23}$$
$$\geq \mathop{\mathbb{E}}_{x\sim U} \alpha_+^2(B_i(x)) \geq (\delta\varepsilon)^2. \tag{24}$$

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

4 Dense model theorem

Definition 4.1: For a set $S \subseteq U$ and a function $T : U \to \{0,1\}$, let $T(S) := \mathbb{E}_{x\in S}T(x)$. (For a measure $\mu : U \to [0,1]$, also write $T(\mu) = \mathbb{E}_{x\sim\mu}T(x)$.)

Let $S \subseteq U$ be a subset, and let $\mathcal{T}$ be a set of tests. S is (ε, δ)**-pseudo-dense against** T 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.

1. One way of being pseudo-dense is to actually be dense.
2. Another, one step removed, is that there's a set R that's indistinguishable from the whole distribution (in the sense of being at least about δ fraction of it), and the set is dense in R.

Definition 4.2: 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 $T \in \mathcal{T}$,

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

Theorem 4.3 (Dense model theorem)**.** *Let $\mathcal{T}$ be a class of tests $U \to \{0,1\}$.*

If S is (ε, δ)-pseudodense against $F_k\mathcal{T}$, $k = O\left(\frac{1}{\varepsilon^2\delta^2}\right)$ 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.

Proof. 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 $T \in \mathcal{F}_k\mathcal{T}$,

$$\mathbb{P}_{U'}[T((y,x)) = y] = \delta' T(S) + (1-\delta')(1 - T(U)) = 1 - \delta' + \delta'(T(S)) - (1-\delta')T(U) \quad (25)$$

$$= 1 - \delta' + \frac{1}{1+\delta}(\delta T(S) - T(U)) \leq 1 - \delta' + \varepsilon. \quad (26)$$

No test in $\mathcal{F}_k\mathcal{T}$ can be correct with probability $> \delta' - \varepsilon$. By the Hard-core Lemma 3.2, there exists $|\mu'| \geq 2(\delta' - \varepsilon)$ such that for any $T \in \mathcal{T}$, $\mathbb{P}_{(x,y)\sim U'}[T(x) = y] \leq \frac{1}{2} + \varepsilon$.

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 $2(\delta' - \varepsilon)\left(\frac{1}{2} - \varepsilon\right) = \delta'\left(1 - O\left(\frac{\varepsilon}{\delta}\right)\right)$ of the mass. Then

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

□