Notes on low-degree likelihood ratio tests

Sidhanth Mohanty

November 15, 2023

1 Prelude: information-computation gaps

Let's start by discussing a few algorithmic problems that have given us a very useful lens into algorithms & complexity in the average case.

Planted k-XOR *aka sparse $\mathbb{F}_2$ linear equations*. In this problem, $\boldsymbol{x} \sim \{\pm 1\}^n$ is a *hidden signal*. We are given m samples $(\boldsymbol{S}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{S}_m, \boldsymbol{y}_m)$ where for each $i \in [m]$:

$$(\boldsymbol{S}_i, \boldsymbol{y}_i): \ \boldsymbol{S} \text{ uniform size-}k \text{ subset of } [n]$$
$$\boldsymbol{y} := \begin{cases} \prod_{i \in \boldsymbol{S}} \boldsymbol{x}_i & \text{with probability } 1 - \varepsilon \\ \text{uniform } \pm 1 \text{ bit} & \text{with probability } \varepsilon \end{cases}$$

Algorithmic task. Given $((\boldsymbol{S}_i, \boldsymbol{y}_i))_{1 \leqslant i \leqslant m}$, recover $\boldsymbol{x}$.

State of affairs. Information theoretic recovery of $\boldsymbol{x}$ known when $m = \Omega(n)$. Efficient algorithms to recover $\boldsymbol{x}$ known when $m = \Omega(n^{k/2})$.

Planted clique in random graph. $\boldsymbol{x}$ is a random k-sparse vector in $\{0,1\}^n$ is a hidden signal aka planted clique. We are given a graph $\boldsymbol{G}$ where:

$$\{i,j\} \text{ edge when } \boldsymbol{x}_i \boldsymbol{x}_j = 1$$
$$\{i,j\} \text{ edge with probability } \frac{1}{2} \text{ when } \boldsymbol{x}_i \boldsymbol{x}_j = 0$$

Algorithmic task. Given $\boldsymbol{G}$, recover the planted clique $\boldsymbol{x}$.

State of affairs. Information theoretic recovery of $\boldsymbol{x}$ known when $m = \Omega(\log n)$. Efficient algorithms to recover $\boldsymbol{x}$ known when $m = \Omega(\sqrt{n})$.

Planted Boolean Vector. $\boldsymbol{x} \sim \{\pm 1\}^n$ is the hidden signal. We are given a uniformly random d-dimensional subspace $\boldsymbol{V}$ of $\mathbb{R}^n$ conditioned on containing $\boldsymbol{x}$.

Algorithmic task. Given $\boldsymbol{V}$, recover the planted Boolean vector $\boldsymbol{x}$.

State of affairs. Information theoretic recovery of $\boldsymbol{x}$ known when $d \leqslant n - 1$. Efficient algorithms for recover known when $d = O(\sqrt{n})$.

Our main motivating question is:

Question 1.1. *How can we obtain rigorous evidence for the hardness in the regime where information-theoretic algorithms exist but no efficient algorithms are known?*

2 On proving hardness for average-case problems

In statistics, the inputs to problems are very structured, often comprising of *independent* samples, or graphs and matrices where entries enjoy a lot of independence. However, as we saw above, despite this nice structure, several algorithmic problems of interest are nevertheless seemingly hard. The theory of NP-hardness fails to articulate why such problems are hard, since reductions from known hard problems do not give such nice structured instances.

Two popular approaches to support that a problem is hard are:

1. **Hardness against restricted models.** For example, lower bounds against *low-degree polynomials*, *convex programming hierarchies*, *message passing algorithms*, *statistical query models*, *local algorithms*, etc.

2. **Average-case reductions.** Instead of starting with an assumption like $P \neq NP$, start with an assumption like "Planted Clique is hard when $k \ll \sqrt{n}$", or "Planted kXOR is hard when $m = 100n$".

In this lecture, we will focus on the first approach to proving hardness, focusing on a model for low-degree polynomial-based algorithms.

3 Problem set-up

Consider the following algorithmic problem, commonly known as *distinguishing* or *hypothesis testing*:

Distinguishing/hypothesis testing. Let $\mathcal{N}$ (*null distribution*) and $\mathcal{P}$ (*planted distribution*) be two probability distributions. Given $\boldsymbol{G}$ drawn from either $\mathcal{N}$ or $\mathcal{P}$, figure out whether the sample came from $\mathcal{N}$ or $\mathcal{P}$.

Think of the planted distribution $\mathcal{P}$ as akin to the three distributions mentioned at the start of lecture, and think of the null distribution $\mathcal{N}$ as being a version of the distribution with no hidden signal. Concretely:

- In the *planted clique* problem, the null distribution is an Erdős–Rényi graph $G(n, 1/2)$ [every pair of vertices ij is independently chosen as an edge with probability $1/2$].
- In the *planted kXOR* problem, in the null distribution, the $\boldsymbol{y}_i$ are all chosen as uniform ± 1 bits.
- In the *planted Boolean vector* problem, the null distribution is a uniformly random d-dimensional subspace of $\mathbb{R}^n$.

Information-theoretic indistinguishability. Our model for impossibility to *efficiently* solve some hypothesis testing problems is based on low-degree polynomials can be motivated by information-theoretic techniques. (This is with the privilege of hindsight — the path to this model was a lot murkier than the story below.)

The distinguisher with the highest success probability that can tell $\mathcal{N}$ and $\mathcal{P}$ apart is the following function.

$$F(\boldsymbol{G}) \coloneqq \begin{cases} \mathcal{N} & \text{when } \mathcal{N}(\boldsymbol{G}) > \mathcal{P}(\boldsymbol{G}) \\ \mathcal{P} & \text{otherwise.} \end{cases}$$

Concretely, this function maximizes

$$\mathbf{Pr}_{\mathcal{N}}[F(\boldsymbol{G}) = \mathcal{N}] - \mathbf{Pr}_{\mathcal{P}}[F(\boldsymbol{G}) = \mathcal{N}] = \mathbf{Pr}_{\mathcal{P}}[F(\boldsymbol{G}) = \mathcal{P}] - \mathbf{Pr}_{\mathcal{N}}[F(\boldsymbol{G}) = \mathcal{P}],$$

whose the value is equal to the *total variation distance*:

$$d_{\mathrm{TV}}(\mathcal{N}, \mathcal{P}) \coloneqq \mathbf{E}_{\boldsymbol{G} \sim \mathcal{N}} \left[\left| 1 - \frac{\mathcal{P}(\boldsymbol{G})}{\mathcal{N}(\boldsymbol{G})} \right| \right].$$

When the TV distance between $\mathcal{N}$ and $\mathcal{P}$ is tiny, it is not possible to reasonably tell these models apart from $\boldsymbol{G}$.

Exercise 3.1. *Prove that the total variation distance is equal the maximum of the above objective.*

In the settings we are studying, the TV distance is close to 1, but it nevertheless seems hard to tell these algorithms apart with an efficient algorithm. The issue is that the distinguishing function F may not be efficiently computable. We would thus like to get a *computational version* of TV distance, that articulates when efficient algorithms cannot tell two distributions apart. In particular, the goal is to get a handle on "something like":

$$\max_{\substack{F:\text{inputs} \to \{\mathcal{N}, \mathcal{P}\} \\ F \text{ efficiently computable}}} \mathbf{Pr}_{\mathcal{N}}[F(\boldsymbol{G}) = \mathcal{N}] - \mathbf{Pr}_{\mathcal{P}}[F(\boldsymbol{G}) = \mathcal{N}].$$

We would like to simplify the above expression for two reasons:

1. Our understanding of the space of "efficiently computable" functions is at a hopeless state.
2. Even if we replaced "efficiently computable" with some nicer set, it is typically not analytically nice to try to maximize over Boolean functions.

One also grapples with the second point while proving information-theoretic lower bounds. A relaxation of the TV distance is the *chi-squared divergence*, denoted $\chi^2(\mathcal{P}\|\mathcal{N})$.

$$d_{\mathrm{TV}}(\mathcal{N},\mathcal{P}) := \mathbf{E}_{\boldsymbol{G}\sim\mathcal{N}}\left[\left|1-\frac{\mathcal{P}(\boldsymbol{G})}{\mathcal{N}(\boldsymbol{G})}\right|\right] \leqslant \sqrt{\mathbf{E}_{\boldsymbol{G}\sim\mathcal{N}}\left[\left(1-\frac{\mathcal{P}(\boldsymbol{G})}{\mathcal{N}(\boldsymbol{G})}\right)^2\right]} =: \chi^2(\mathcal{P}\|\mathcal{N}).$$

In many scenarios, it is a much easier quantity to control analytically. An alternate "variational" form for the chi-squared divergence[1] is the following:

$$\chi^2(\mathcal{P}\|\mathcal{N}) = \max_{F:\text{inputs}\to\mathbb{R},F\neq 0} \frac{\mathbf{E}_{\mathcal{P}}F - \mathbf{E}_{\mathcal{N}}F}{\sqrt{\mathbf{Var}_{\mathcal{N}}F}}$$

Exercise 3.2. *Prove the above variational formula for chi-squared divergence. Show that the optimizer to the problem on the right is achieved by choosing $F(\boldsymbol{G})$ as $\frac{\mathcal{P}(\boldsymbol{G})}{\mathcal{N}(\boldsymbol{G})}-1$.*

The second relaxation that we make is to replace "efficiently computable" with an expressive class of functions that are also analytically nice to get a handle on — low-degree polynomials! This motivates defining

$$\chi^2_{\leqslant D}(\mathcal{P}\|\mathcal{N}) = \max_{\substack{F:\text{inputs}\to\mathbb{R},F\neq 0\\ F\text{ degree-}\leqslant D\text{ polynomial}}} \frac{\mathbf{E}_{\mathcal{P}}F - \mathbf{E}_{\mathcal{N}}F}{\sqrt{\mathbf{Var}_{\mathcal{N}}F}} \tag{1}$$

which we call the *degree-D* chi-squared divergence; this is more popularly known as the *low-degree likelihood ratio* for reasons we will see soon.

The hardness hypothesis surrounding the low-degree chi-squared divergence is that it is a good proxy for computational indistinguishability. Concretely, as first posited in [HS17]:

> Suppose $\mathcal{N}$ and $\mathcal{P}$ are sufficiently "well-structured" distributions over $\mathbb{R}^n$,[2] and $\chi^2_D(\mathcal{P}\|\mathcal{N}) = o_n(1)$, then there is no $n^{O(D/\log n)}$-time algorithm to distinguish $\mathcal{N}$ from $\mathcal{P}$.

Power of the low-degree model. The power of the low-degree model comes from the fact that it is quite easy to obtain a handle on $\chi^2_{\leqslant D}(\mathcal{P}\|\mathcal{N})$ for many distributions of interest, and thus predict where the computational threshold lies. The predictions obtained from this method also accurately line up with when many of our algorithmic techniques fail for several problems of interest.

It is possible to exactly characterize the function F that achieves the maximum in the definition of the low-degree divergence from Eq. (1), which enables explicitly bounding the divergence relatively painlessly.

Theorem 3.3. *The function F achieving the maximum in Eq. (1) is given by:*

$$F(\boldsymbol{G}) = \left(\frac{\mathcal{P}}{\mathcal{N}}\right)^{\leqslant D}(\boldsymbol{G}) - 1 \qquad \chi^2_D(\mathcal{P}\|\mathcal{N}) = \left\|\left(\frac{\mathcal{P}}{\mathcal{N}}\right)^{\leqslant D}(\boldsymbol{G}) - 1\right\|_{\mathcal{N}}$$

Here, for a function H, the notation $H^{\leqslant D}$ refers to its projection onto the subspace of degree-$\leqslant D$ polynomials. The projection and norm $\|\cdot\|_{\mathcal{N}}$ is under the inner product $\langle H_1, H_2\rangle_{\mathcal{N}} = \mathbf{E}_{\boldsymbol{G}\sim\mathcal{N}} H_1(\boldsymbol{G})\cdot H_2(\boldsymbol{G})$.

[1] Variational means it arises as the solution to some optimization problem.

[2] see [Hop18] for a more precise formulation

We now derive the formula for F and $\chi^2_D(\mathcal{P}\|\mathcal{N})$.

Proof of Theorem 3.3. Observe that the objective in Eq. (1) is invariant to shifting and rescaling F, and thus, we can write our optimization problem as:

$$\max_{\substack{F:\deg(F)\leqslant D\\ \mathbf{E}_{\mathcal{N}}F(\boldsymbol{G})=0\\ \mathbf{E}_{\mathcal{N}}F(\boldsymbol{G})^2=1}} \mathbf{E}_{\mathcal{P}}F(\boldsymbol{x}) = \max_{\substack{F:\deg(F)\leqslant D\\ \mathbf{E}_{\mathcal{N}}F(\boldsymbol{G})=0\\ \mathbf{E}_{\mathcal{N}}F(\boldsymbol{G})^2=1}} \mathbf{E}_{\mathcal{N}}\left[F(\boldsymbol{G})\cdot\frac{\mathcal{P}(\boldsymbol{G})}{\mathcal{N}(\boldsymbol{G})}\right]$$

Now we use the following general linear algebra fact:

Fact 3.4. *Let V be a subspace of an inner product space $\mathcal{H}$, and let Π_V be the orthogonal projection onto V. Π_V is self-adjoint, and consequently if $x \in V$ and $y \in \mathcal{H}$, then*

$$\langle x, y\rangle_{\mathcal{H}} = \langle \Pi_V x, y\rangle_{\mathcal{H}} = \langle x, \Pi_V y\rangle_{\mathcal{H}}.$$

By setting $\mathcal{H}$ as the space of (measurable) functions from $\mathbb{R}^d$ to $\mathbb{R}$ equipped with inner product

$$\langle f, g\rangle_{\mathcal{N}} := \mathbf{E}_{\boldsymbol{G}\sim\mathcal{N}} f(\boldsymbol{G})g(\boldsymbol{G})$$

and V as $\{F : \deg(F) \leqslant D,\ \mathbf{E}[F] = 0\}$ and applying Fact 3.4, we get

$$\begin{aligned}
F^*_D &= \arg\max_{\substack{F\in V\\ \langle F(\boldsymbol{x}),F(\boldsymbol{x})\rangle_{\mathcal{N}}=1}} \left\langle F, \left(\frac{\mathcal{P}}{\mathcal{N}}\right)^{\leqslant D} - \mathbf{E}_{\boldsymbol{x}\sim\mathcal{N}}\frac{\mathcal{P}(\boldsymbol{x})}{\mathcal{N}(\boldsymbol{x})}\right\rangle_{\mathcal{N}}\\
&= \arg\max_{\substack{F\in V\\ \langle F(\boldsymbol{G}),F(\boldsymbol{G})\rangle_{\mathcal{N}}=1}} \left\langle F, \left(\frac{\mathcal{P}}{\mathcal{N}}\right)^{\leqslant D} - 1\right\rangle_{\mathcal{N}}\\
&= \frac{\left(\frac{\mathcal{P}}{\mathcal{N}}\right)^{\leqslant D} - 1}{\left\|\left(\frac{\mathcal{P}}{\mathcal{N}}\right)^{\leqslant D} - 1\right\|_{\mathcal{N}}}.
\end{aligned}$$

Finally,

$$\mathbf{E}_{\mathcal{P}}F^*_D(\boldsymbol{G}) = \left\langle F^*_D, \frac{\mathcal{P}}{\mathcal{N}}\right\rangle = \left\|\left(\frac{\mathcal{P}}{\mathcal{N}}\right)^{\leqslant D} - 1\right\|_{\mathcal{N}}.$$ □

4 Case of planted clique

Simple algorithm when $k \geqslant 10\sqrt{n\log n}$. In $G(n, 1/2)$, every vertex has degree $n/2 \pm 5\sqrt{n\log n}$ with high probability. When $k \geqslant 10\sqrt{n\log n}$, all the clique vertices systematically have larger degree. Distinguisher can simply be based on the maximum degree vertex. Clique can be recovered by taking top-k vertices, sorted in decreasing order of degree.

There is a spectral algorithm to recover a clique of size $O(\sqrt{n})$ in polynomial time.

Exercise 4.1. *For every constant $\alpha > 0$, give an algorithm that runs in time $n^{O(\log\frac{1}{\alpha})}$ to find the planted clique when $k \geqslant \alpha\sqrt{n}$.*

Intractability We will now see how the low-degree method predicts the $\sqrt{n}$ threshold for planted clique.

Let $\mathcal{N}$ be $G(n,1/2)$ and $\mathcal{P}$ be the planted clique distribution with clique size k. Our goal in this section is to compute $\chi^2_D(\mathcal{P}\|\mathcal{N})$ for $D = O(1)$ as a function of k. Using $\mathsf{LR}(\boldsymbol{G})$ to denote $\frac{\mathcal{P}}{\mathcal{N}}(\boldsymbol{G})$, and Parseval's identity from Fourier analysis:

$$\chi^2_D(\mathcal{P}\|\mathcal{N})^2 = \|\mathsf{LR}^{\leqslant D} - 1\|^2_{\mathcal{N}} = \sum_{\alpha \subseteq \binom{[n]}{2}} \langle \mathsf{LR}^{\leqslant D} - 1, \chi_\alpha \rangle^2$$

where χ_α are the Fourier characters of functions from $\{\pm 1\}^{\binom{[n]}{2}}$ to $\mathbb{C}$.

Towards computing the Fourier coefficients, fix $\alpha \subseteq \binom{[n]}{2}$. If $|\alpha| > D$, $\langle \mathsf{LR}^{\leqslant D}, \chi_\alpha \rangle = 0$. And if $1 \leqslant |\alpha| \leqslant D$, then

$$\begin{aligned}
\langle \mathsf{LR}^{\leqslant D} - 1, \chi_\alpha \rangle &= \langle \mathsf{LR}, \chi_\alpha \rangle \\
&= \mathbf{E}_{\mathcal{P}} \chi_\alpha(\boldsymbol{G}) \\
&= \mathbf{E}_{\mathcal{P}} \prod_{\{i,j\} \in \alpha} \boldsymbol{G}_{ij} \\
&= \mathbf{Pr}[\text{all vtcs touched by } \alpha \text{ in clique}] \cdot \mathbf{E}_{\mathcal{P}} \left[\prod_{\{i,j\} \in \alpha} \boldsymbol{G}_{ij} \mid \text{all vtcs touched by } \alpha \text{ in clique} \right] \\
&= \left(\frac{k}{n}\right)^{|V(\alpha)|}
\end{aligned}$$

Thus:

$$\begin{aligned}
\chi^2_D(\mathcal{P}\|\mathcal{N})^2 &= \sum_{1 \leqslant |\alpha| \leqslant D} \left(\frac{k}{n}\right)^{2|V(\alpha)|} \\
&\leqslant \sum_{t \leqslant 2D} 2^{\binom{t}{2}} n^t \left(\frac{k}{n}\right)^{2t} \\
&\leqslant C \left(\frac{k^2}{n}\right)^{2D}
\end{aligned}$$

where $C = O(1)$ because $D = O(1)$.

Observe that this is $o(1)$ whenever $k = o(\sqrt{n})$!

Exercise 4.2. *For $\mathcal{N} = G(n,1/2)$ and $\mathcal{P} = G(n,1/2) + k$-clique, for $k = o(\sqrt{n})$ and $D = o(\log^2 n)$, prove that $\chi^2_D(\mathcal{P}\|\mathcal{N}) = o_n(1)$.*

Exercise 4.3. *For $\mathcal{N}$ = random kXOR and $\mathcal{P}$ = planted kXOR where number of variables is n and number of constraints is m, and $k \geqslant 3$, prove that for $D = o\left(\left(\frac{m}{n}\right)^{2/(k-1)}\right)$, prove that $\chi^2_D(\mathcal{P}\|\mathcal{N}) = o_n(1)$.*

Exercise 4.4 (Hard). *For $\mathcal{N}$ = random d-dimensional subspace of $\mathbb{R}^n$ and $\mathcal{P}$ =planted Boolean vector distribution with same parameters, identify the (n,d,D) choices for which $\chi^2_D(\mathcal{P}\|\mathcal{N}) = o_n(1)$.*

5 Connections and broader discussion

Since the quantity $\chi^2_D(\mathcal{P}\|\mathcal{N})$ is quite easy to get a handle on and perform explicit calculations with for many choices of $\mathcal{N}$ and $\mathcal{P}$, it is a very appealing predictor for computational thresholds.

However, it is not known to imply lower bounds against any actual classes of algorithms. Here, we state a few problems of interest.

- **Lower bounds against low-degree algorithms.** Show that $\chi^2_D(\mathcal{P}\|\mathcal{N}) = o_n(1)$ implies a lower bound against algorithms that: (1) evaluate a polynomial p on the input $\boldsymbol{G}$, (2) threshold on the value of $p(\boldsymbol{G})$ to output $\mathcal{P}$ or $\mathcal{N}$.
- **Lower bounds against spectral algorithms.** Show that if $\chi^2_{\mathrm{poly}(D,\log n)}(\mathcal{P}\|\mathcal{N}) = o_n(1)$, then for any $n^D \times n^D$ matrix $M(\boldsymbol{G})$ constructed by placing a degree-D polynomial of $\boldsymbol{G}$ in every entry of $M(\boldsymbol{G})$, $\mathrm{spectrum}(M(\boldsymbol{G}))_{\boldsymbol{G}\sim\mathcal{N}}$ is "close" to $\mathrm{spectrum}(M(\boldsymbol{G}))_{\boldsymbol{G}\sim\mathcal{P}}$. (A quantitatively strong version of this would imply lower bounds against the Sum-of-Squares hierarchy!)

Some highlights related to other models that low-degree lower bounds are related to.

- A heuristic from statistical physics, called the cavity method, was used to obtain predictions for the algorithmic threshold for *community detection* in *stochastic block models* in [DKMZ11]. The paper where the low-degree divergence was introduced [HS17] proved that the low-degree threshold for the block model matches the cavity method predictions.
- Under fairly general conditions on $\mathcal{N}$ and $\mathcal{P}$, the work of [BBH+20] proved that the low-degree threshold matches the threshold for the *statistical query model*, a restrictive query model introduced in the context of learning theory.
- For a class of $\mathcal{N}$ and $\mathcal{P}$, "Gaussian additive models", the work of [BEAH+22] showed that the low-degree threshold matches the one predicted by certain solution geometry-based techniques from statistical physics.
- A similar theory was developed for *estimation* problems in the work of Schramm & Wein [SW22].

References

[BBH+20] Matthew Brennan, Guy Bresler, Samuel B Hopkins, Jerry Li, and Tselil Schramm. Statistical query algorithms and low-degree tests are almost equivalent. *arXiv preprint arXiv:2009.06107*, 2020. 7

[BEAH+22] Afonso S Bandeira, Ahmed El Alaoui, Samuel Hopkins, Tselil Schramm, Alexander S Wein, and Ilias Zadik. The franz-parisi criterion and computational trade-offs in high dimensional statistics. *Advances in Neural Information Processing Systems*, 35:33831–33844, 2022. 7

[DKMZ11] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6):065701, 2011. 7

[Hop18] Samuel Hopkins. *Statistical inference and the sum of squares method*. PhD thesis, Cornell University, 2018. 4

[HS17] Samuel B. Hopkins and David Steurer. Efficient Bayesian Estimation from Few Samples: Community Detection and Related Problems. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 379–390, 2017. 4, 7

[SW22] Tselil Schramm and Alexander S Wein. Computational barriers to estimation from low-degree polynomials. *The Annals of Statistics*, 50(3):1833–1858, 2022. 7