Journal of Machine Learning Research 1 (2000) 1-48 Submitted 4/00; Published 10/00

Review: Planted Clique as a Case Study for Statistical-Computational Gaps

William Merrill WILLM@NYU.EDU
Center for Data Science
New York University

Nikolaos Tsilivis NT2231@NYU.EDU
Center for Data Science
New York University

Editor: No editor

Abstract

This survey covers the recent research area of statistical-computational gaps through the lens of the planted clique problem. Statistical problems are said to exhibit such a gap if there is a complexity threshold where the problem is statistically possible to solve, but no efficient algorithm can solve the problem. We first introduce statistical-computational gaps through the canonical example of the well-studied planted clique problem. We review the analysis of when the problem is statistically possible to solve. We next discuss efficient algorithms that solve the problem for lower-complexity instances, motivating the conjecture that planted clique exhibits a statistical-computational gap. We then turn towards recent work aiming to establish and predict the gap for planted clique, first for low-degree polynomial algorithms, and then for statistical query algorithms. Overall, our review presents a cohesive treatment of the planted clique problem as a case study for understanding statistical-computational gaps, and covers recent work analyzing planted clique that may lead towards a broader theory for predicting statistical-computational in other problems as well.

Keywords: Statistical-computational gaps, planted clique, statistical queries, low-degree polynomials

1. Introduction

Unconventionally, this survey includes two introductions, one from the perspective of statistics and one from the angle of computer science.

In 1935, Ronald Fisher, provoked by the claims of Muriel Bristol that she could tell by tasting warm beverages whether the milk has been added to the cup before or after the tea, came up with the concept of *hypothesis testing*. Observations are collected from the world (which is believed to be governed by some randomness) and the task of a statistician is to design a decision rule that decides between two possible hypotheses, or as stated originally, that "possibly disproves the null hypothesis". The null hypothesis is the one of the two that is the most "likely" to occur from chance alone. The main statistical question that arises here is: how many cups of tea do we need to provide to Mrs. Bristol to determine with high confidence whether she has indeed tea-tasting super capabilities or not, that is

how much information do we need for solving a statistical problem?

©2000 Merrill and Tsilivis.

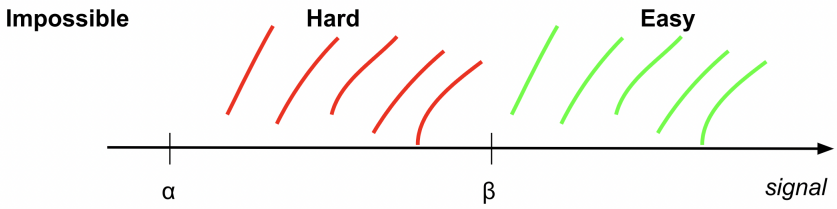


Figure 1: The power of signal in the data is conjectured to induce 3 different regimes in statistical inference tasks: (a) An **Impossible** one, where there is simply not enough information to statistically solve a problem, (b) An **Easy** one, where the power of the signal is sufficient to do inference with an efficient (polynomial time) algorithm, and (conjecturally) (c) A **Hard** one, where the problem is statistically possible to be solved, but there is no polynomial time algorithm that does so.

A following question is *what is this decision rule that solves the problem* and, perhaps more crucially in the recent era of automated computation,

what is the computational complexity of the decision rule that solves a statistical problem?

The phenomenon of *computational-statistical gaps*, which is by no means limited to hypothesis testing problems[1], but can be observed in many statistical inference tasks, is the fascinating observation that optimal answers to the previous two questions might be at tension with each other; a procedure that is time efficient (takes polynomial time to terminate) might require much more statistical information than what is required for solving it with any algorithm (see Figure 1).

In the early 1970's, Richard Karp organised the blooming field of computational complexity by showing that there exist a number (21) of different problems which are at least as hard as the one of satisfying boolean formulae. These problems are believed to be computationally hard, in the sense that there exist *worst-case* instances for which any algorithm that solves the problems on these instances must pay superpolynomial time. Famously, this consideration has limitations as a worst case analysis might be too pessimistic for the real world (Roughgarden, 2021), and one approach that attempts to go beyond this is the *average* case approach, where instances are assumed to be random variables that come from specific distributions. While this area is far from developed in the level of traditional Computational Complexity, there is gradual progress on which problems are *hard* here, and, interestingly, in what regimes. At the heart of this endeavor for an average case Computational Complexity theory seems to lie the phenomenon of *statistical-computational gaps*, which informally demonstrates that certain problems might be either easy or hard, depending on the "size" of the pattern that we are searching for inside a random input (see Figure 1).

The rest of this survey is organised as follows. Section 2 fixes some notation. Section 3 introduces the phenomenon of statistical-computational gaps through a problem that is considered to be canonical in this area, the planted clique problem. Sections 4 and 5

1. however, this survey will focus on one such problem, the planted clique problem.

cover different classes of efficient algorithms, for which there exist lower bounds on the statistical information needed for them to succeed in inferring the statistical truth; that is, the conjectured gaps hold for these special classes of algorithms. In particular, Section 4 covers statistical query algorithms and Section 5 covers low degree polynomials.

Perspective and Prior Work. The purpose of this survey is to give a short, pedagogical introduction to the area of statistical-computational gaps, particularly aimed at young researchers/graduate students in the fields of Computer Science, Data Science and Statistics, through the problem of Planted Clique. There is a number of excellent different references that cover some aspects of this area already, and we have mainly based our exposition on the works of Kunisky et al. (2022); Ilias Zadik (2019); Zdeborová and Krzakala (2016).

2. Notation

We will use log to denote the logarithm in base 2, following conventions in computer science. Whenever we refer to an undirected graph going forward, we mean an undirected graph with no self-loops.

3. The Gap Conjecture for Planted Clique

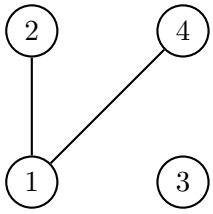
It is most concrete to introduce the idea of a statistical-computational gap through a canonical example of a problem that is thought to exhibit one. The planted clique problem is such in example (Jerrum, 1992; Gamarnik and Zadik, 2019; Chen et al., 2022). At a high level, this problem consists of finding the k clique in a random undirected graph of n nodes. Whereas it is statistically possible to detect the planted clique when $k \geq (2+\epsilon)\log n$ for some small $\epsilon > 0$ (Chen et al., 2022), there is only known to be a polynomial-time algorithm for detection when $k \geq \Omega(\sqrt{n})$. Thus, when $(2+\epsilon)\log n < k < \Omega(\sqrt{n})$, the planted clique problem is statistically possible to solve, but there is no known efficient algorithm for doing so.

This motivates the conjecture that there is a *statistical-computational gap* for planted clique, i.e., that any efficient algorithm to find the planted clique must require a larger value of k than the value necessary to statistically distinguish the planted clique from other cliques. Yet, this gap remains a conjecture, since there could be an unknown algorithm that solves planted clique efficiently for values of k smaller than $\Omega(\sqrt{n})$. Recent work attempting to understand statistically-computational gaps has proven such an algorithm cannot exist, at least for restricted classes of statistical algorithms. After introducing planted clique in more detail in this section, we will turn to discussing this recent work.

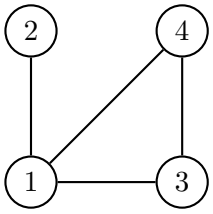
3.1 Problem Definition

We now define the planted clique problem more formally. There are really two related variants of planted clique that we will discuss: *recovery* and *detection*.

Recovery. In the recovery variant planted clique, we first sample a random undirected graph by creating n nodes, and choosing whether to connect each pair of nodes with probability $\frac{1}{2}$, as showin in the left side of Figure 2. We denote the distribution of this initial



(a) Example initial graph ($n = 4$) with probability $2^{-\binom{4}{2}} = \frac{1}{64}$ under $\mathcal{G}(4, \frac{1}{2})$.



(b) Resulting graph after planting a 3-clique at nodes $\{1, 3, 4\}$.

Figure 2: Example of (a) a random graph sampled from $\mathcal{G}(4, \frac{1}{2})$ and (b) a possible resulting graph under $\mathcal{G}(4, \frac{1}{2}, 3)$ with a 3-clique planted at nodes $\{1, 3, 4\}$.

graph $\mathcal{G}(n, \frac{1}{2})$. We then plant a clique in this graph, i.e., we choose k out of n edges to fully connect. We denote the distribution of this new graph, which is guaranteed to contain a k-clique, as $\mathcal{G}(n, \frac{1}{2}, k)$. The planted clique problem takes as input $G \sim \mathcal{G}(n, \frac{1}{2}, k)$ and expects as output the indices of the k nodes that form the planted clique that we injected into G.

Detection. The planted clique problem can be equivalently stated with a more statistical flavor as follows. We imagine we are given a graph G coming from either $\mathcal{G}(n, \frac{1}{2})$ or $\mathcal{G}(n, \frac{1}{2}, k)$, and we wish to distinguish the distributions, i.e., test the *null* hypothesis that G came from $\mathcal{G}(n, \frac{1}{2})$. The detection variant effectively reduces to distinguishing whether a k-clique found in G was planted or occurred by chance, and is thus "at most as hard as" the recovery version of planted clique.

Formally, the task of *hypothesis testing* can be formulated as follows. Let $\underbrace{\mathcal{Q} = (\mathcal{Q}_n)_{n\in\mathbb{N}}}_{\textit{null hypothesis}}$ and $\underbrace{\mathcal{P} = (\mathcal{P}_n)_{n\in\mathbb{N}}}_{\textit{planted hypothesis}}$ be two (sequences of) probability distributions over a (sequence of) measurable space(s) $\mathcal{X} = (\mathcal{X}_n)_{n\in\mathbb{N}}$, where we overload notation to also let $\mathcal{X}$ refer to the union of these spaces. One can think of n as the size of the problem. The goal of hypothesis testing is to design a *statistic* or *test* f_n that, given data $\mathbf{X} \in \mathcal{X}_n$ for some n, decides the "true" generating distribution with high probability as the size of the instance gets bigger and bigger. That is, we search for an $f_n : \mathcal{X}_n \to \{0, 1\}$ such that:

$$\mathbb{P}_{\mathbf{X}\sim\mathcal{P}_n}\{f_n(\mathbf{X}) = 1\} \xrightarrow{n\to\infty} 1$$

and

$$\mathbb{P}_{\mathbf{X}\sim\mathcal{Q}_n}\{f_n(\mathbf{X}) = 0\} \xrightarrow{n\to\infty} 1.$$

In words, the test must with high probability output 1 when the data come from the planted distribution and 0 otherwise. In that case, we will say that f_n *distinguishes* between $\mathcal{P}$ and $\mathcal{Q}$. As noted, this definition is concerned with what happens in the asymptotic limit $n \to \infty$.

3.2 Statistical Possibility

The first question we seek to answer about planted clique when it is statistically possible to find the planted clique. Informally, when k is small, there may randomly be other cliques

of size k in the random graph, which forces us to guess which one was planted. Thus, we cannot identify the planted clique with high probability. As k increases, it becomes vanishingly less likely that some set of nodes should form a k-clique in the random graph, as all k nodes will have to share an edge, and each edge has probability $1/2$. We thus have the intuition that the problem may not be statistically feasible for smaller values of k, but that it becomes so as some larger threshold for k.

We will seek to identify the minimum value of k for which it is possible to identify the planted clique. Intuitively, this is statistically possible when the probability that there is another k clique in the original graph is small. It turns out that, which high probability, the maximum clique in a graph of n nodes has size $(2+\epsilon)\log n$, for some small positive ϵ. Thus, as long as $k \geq (2+\epsilon)\log n$, there is a high probability that the planted clique is the only k clique, and the statistical estimation problem is thus possible.

We have found that most presentations of planted clique in the literature do *not* justify that it is unlikely to have a maximum clique greater than $2\log n$. In the interest of completeness, we provide such a proof, adapted from the rough proof sketch given by Trevisan (2017):

Lemma 1 *With probability at least* $1 - \frac{1}{n^{\Omega(1)}}$*, the maximum clique in a graph* $G \sim \mathcal{G}(n, \frac{1}{2})$ *has size at most* $2\log n + 2$*.*

Proof Let the random variable N_k be the number of k-cliques in $G \sim \mathcal{G}(n, \frac{1}{2})$. Then,

$$\begin{aligned}\mathbb{E}[N_k] &= \binom{n}{k} \cdot \left(\frac{1}{2}\right)^{\binom{k}{2}} \\ &\leq 2^{k\log n + \frac{k}{2} - \frac{k^2}{2}} \\ &= 2^{-\frac{k}{2}(k-1-2\log n)}.\end{aligned}$$

For all $k \geq 2\log n + 2$, this becomes

$$\begin{aligned}\mathbb{E}[N_k] &\leq 2^{-\frac{2\log n+2}{2}(2\log n+2-1-2\log n)} \\ &= 2^{-\log n - 1} \leq \frac{1}{n^{\Omega(1)}}.\end{aligned}$$

By applying Markov's inequality, we get

$$\begin{aligned}\Pr[N_k \geq 1] &\leq \frac{\mathbb{E}[N_k]}{1} \\ &= \frac{1}{n^{\Omega(1)}}.\end{aligned}$$

Let N be the number of cliques in G of size at least $2\log n + 2$. Because any clique of size larger than $2\log n + 2$ would contain a clique of size $2\log n + 2$, we get that

$$\begin{aligned}\Pr[N \geq 1] &\leq \Pr[N_{2\log n+2} \geq 1] \\ &\leq 2^{-\Omega(\log n)} \\ \therefore \Pr[N = 0] &\geq 1 - \frac{1}{n^{\Omega(1)}}.\end{aligned}$$

■

Note that tight concentration can also be shown in the other direction: i.e., this is not just an upper bound, but the maximum clique in G is typically around $2 \log n$ (Trevisan, 2017). However, for our purposes, the upper bound is sufficient to demonstrate that it is very unlikely to observe a clique of size much greater than $2 \log n$ in a large initial graph.

To recap, Lemma 1 is the core graph-theoretic tool that allows us to analyze whether the planted clique is detectable. As long as $k \geq (2+\epsilon) \log n$, for large n we have $k \geq 2 \log n + 2$. It is thus very unlikely to find a "natural" k-clique in the original graph, so any clique we do find is probably planted. We conclude that, with high probability, we can uniquely identify the planted clique, or, equivalently, identify which distribution the graph came from.

3.3 Hypothesis Testing Algorithm

We now discuss an algorithm for the hypothesis testing variant of planted clique when $k \geq \Omega(\sqrt{n})$, following the textbook presentation in Arora and Barak (2009).

The test procedure is motivated by understanding the distribution of the vertex degree of a node in G. The Bernoulli random variable E describing whether a pair of nodes have an edge has the following mean and variance:

$$\mathbb{E}[E] = \frac{1}{2}$$
$$\text{Var}[E] = \left(1 - \frac{1}{2}\right)^2 + \left(0 - \frac{1}{2}\right)^2 = \frac{1}{2}.$$

The vertex degree is then simply a sum of $n-1$ independent Bernoulli variables following this distribution. By the central limit theorem, for large n, the vertex degree is approximately Gaussian distributed with mean $\frac{n-1}{2}$ and standard deviation $\frac{\sqrt{n-1}}{4}$. Since we are in the asymptotic regime of large n, we can simplify this to a mean of $\frac{n}{2}$ and variance $\frac{\sqrt{n}}{2}$.

How does this analysis help us with planted clique? If we observe a k-clique with $k = \Omega(\sqrt{n})$, then there is roughly a perturbation from the null distribution equal to one standard deviation, hence detectable. This is the basic idea of a test that can distinguish between the 2 hypotheses.

3.4 Other Canonical Problems with Gaps

We have chosen to introduce statistical-computational gaps by analyzing the planted clique problem because it is perhaps the longest studied and best understood problem thought to exhibit a statistical-computational gap. However, other problems have been analyzed in the literature, including high dimensional linear regression, sparse PCA, tensor PCA and the stochastic block model.

3.5 Summary

The planted clique problem is a canonical example of a problem thought to exhibit a statistical-computational gap. We have discussed how the problem becomes statistically solvable when $k \geq (2+\epsilon) \log n$, but there is no known efficient algorithm to solve it until

$k \geq \Omega(\sqrt{n})$. While the efficient algorithm we provided required $k \geq \Omega(\sqrt{n})$, we have not proven there is not *another* algorithm that is able to solve planted clique for smaller values of k. In general, proving the hardness of planted clique remains open, but progress has been made analyzing restricted classes of hypothesis testing algorithms. In the remainder of the power, we turn to work that takes steps towards establishing lower bounds on the computational hardness of solving planted clique with $k \leq o(\sqrt{n})$.

4. Low-Degree Likelihood Ratio

First we introduce the idea of low-degree likelihood ratio (through the classical in statistics notion of likelihood ratio), and then demonstrate how this method can predict the phase transition from the easy to the hard regime in the planted clique problem.

In classical statistics, an optimal, in a sense which is to be clarified, test for hypothesis testing is the *likelihood ratio* test. The likelihood ratio is simply defined as the ratio between the probability of the data under the planted distribution and the probability of the data under the null distribution:

$$L(\mathbf{X}) = \frac{\mathbb{P}_{\mathbf{X}\sim\mathcal{P}}[\mathbf{X}]}{\mathbb{P}_{\mathbf{X}\sim\mathcal{Q}}[\mathbf{X}]}. \tag{1}$$

The likelihood ratio **test** chooses a threshold η and decides 1 (planted hypothesis) whenever $L(\mathbf{X}) > \eta$, and 0 otherwise.

To see the optimality of the likelihood ratio, it is instructive to define an inner product space over functions. Specifically, let $f, g : \mathcal{X} \to \mathbb{R}$, then:

$$\langle f, g\rangle = \mathbb{E}_{\mathbf{X}\sim\mathcal{Q}}[f(Y)g(Y)]. \tag{2}$$

This inner product measures the correlation of f and g under the null distribution. In this space, we measure the length of a function with the norm: $\|f\| = \sqrt{\mathbb{E}_{\mathbf{X}\sim\mathcal{Q}}[f^2(Y)]}$. Denote by $\mathcal{L}^2(\mathcal{Q})$ the (Hilbert) space of functions for which $\|f\| < \infty$. Under these definitions, the likelihood ratio can be seen as the solution to the following optimization problem over $\mathcal{L}^2(\mathcal{Q})$:

$$\max_{f\in\mathcal{L}^2(\mathcal{Q})} \mathbb{E}_{\mathbf{X}\sim\mathcal{P}}[f(\mathbf{X})],$$
$$\text{s.t. } \|f\| = c,\ c \in \mathbb{R}.$$

By observing that $\mathbb{E}_{\mathbf{X}\sim\mathcal{P}}[f(\mathbf{X})] = \mathbb{E}_{\mathbf{X}\sim\mathcal{Q}}[L(\mathbf{X})f(\mathbf{X})] = \langle L, f\rangle$, and Cauchy-Schwarz inequality, we can indeed see that (a scalar multiple of) the likelihood ratio is indeed optimal. The above problem yields that function f which, under the planted distribution, has in expectation maximum value, while remaining bounded (in norm) under the null distribution.

Now, let us consider again the set up of Section 3.1, where there is a specific dependence of $\mathcal{P}, \mathcal{Q}$ on n, and define $L_n(\mathbf{X}) = \frac{\mathbb{P}_{\mathbf{X}\sim\mathcal{P}_n}\{\mathbf{X}\}}{\mathbb{P}_{\mathbf{X}\sim\mathcal{Q}_n}\{\mathbf{X}\}}$ for any n. That is, we consider a sequence of likelihood ratios. Then, the following fundamental proposition holds.

Proposition 2 *If $\|L_n\|^2$ remains bounded as $n \to \infty$, then there is no test f_n that distinguishes between $\mathcal{P}$ and $\mathcal{Q}$.*

This is often called the *second moment method* in the literature of hypothesis testing, and is an often used tool for tackling hypothesis testing problems.

4.1 Low degree polynomials

Let us focus now on the subspace of $\mathcal{L}^2(\mathcal{Q})$ consisting of all degree (at most) D polynomials, denoted by $\mathcal{V}_D$. This is a proper (linear) subspace. If we consider the constrained optimization problem

$$\begin{aligned} \max_{f \in \mathcal{L}^2(\mathcal{Q})} \ & \mathbb{E}_{\mathbf{X} \sim \mathcal{P}}[f(\mathbf{X})], \\ \text{s.t. } & \|f\| = c, \ c \in \mathbb{R}, \\ & f \in \mathcal{V}_D, \end{aligned}$$

then its solution is simply the projection of the likelihood ratio on $\mathcal{V}_D$, called low degree likelihood ratio and denoted by L_n^D.

The reason why this an important object of study is because it conjecturally captures the essence of computationally efficient hypothesis testing.

Conjecture 2.1 *If $\exists \epsilon > 0$ such that $\|L_n^D\|$ remains bounded for some $D \geq (\log n)^{1+\epsilon}$ as $n \to \infty$, then there is no test f_n computable in polynomial time that distinguishes between $\mathcal{P}$ and $\mathcal{Q}$.*

Notice the similarity with Proposition 2.

4.2 Calculations using the Low Degree method

Let us demonstrate now how the low degree method works in practice. Specifically, we will compute L_n^D for a slight modification of the planted clique problem, and show that when $k = \Theta(\sqrt{n})$, $\|L_n^D\|$ undergoes a phase transition.

We can view the distribution over graphs with n vertices, as a distribution over $\binom{n}{2}$ binary variables. Each variable corresponds to the existence or not of an edge, with $+1$ meaning the edge exists, and -1 meaning that the edge does not exist.

The null distribution $\mathcal{Q}$ is the uniform over $\{\pm 1\}^{\binom{n}{2}}$, and the planted distribution is constructed as follows: each vertex is added to a set C with probability $\frac{n}{k}$, then all edges are added to this set, and for the remaining graph each edge is added with probability $\frac{1}{2}$. This is close to the original planted clique problem, with a size of clique roughly k.

The calculation of $\|L_n^D\|$ can be done through an orthonormal basis over the boolean cube for the polynomials, which in our case is the set of functions

$$h_\alpha(\mathbf{X}) = \prod_{e \in \alpha} \mathbf{X}_e \tag{3}$$

for all $\alpha \subset \binom{n}{2}$. We have then:

$$L_n = \sum_\alpha \langle L_n, h_\alpha \rangle h_\alpha(\mathbf{X}) \tag{4}$$

and the projection on the space of low degree polynomials is

$$L_n^D = \sum_{|\alpha| \leq D} \langle L_n, h_\alpha \rangle h_\alpha(\mathbf{X}), \tag{5}$$

with norm

$$\|L_n^D\|^2 = \sum_{|\alpha| \le D} \langle L_n, h_\alpha \rangle^2. \tag{6}$$

Each coefficient can be calculated through the planted distribution, since

$$\begin{aligned} \langle L_n, h_\alpha \rangle &= \mathbb{E}_{\mathbf{X} \sim \mathcal{Q}}[L_n(\mathbf{X}) h_\alpha(\mathbf{X})] \\ &= \mathbb{E}_{\mathbf{X} \sim \mathcal{P}}[h_\alpha(\mathbf{X})] \\ &= \mathbb{E}_C \mathbb{E}_{\mathbf{X} \sim \mathcal{P}|C}[\prod_{e \in \alpha} \mathbf{X}_e] \\ &= \mathbb{E}_C[\mathbb{1}\{\text{vertices}(\alpha) \subset C\}] \\ &= \mathbb{P}[\text{vertices}(\alpha) \subset C] = \left(\frac{k}{n}\right)^{|\text{vertices}(\alpha)|}. \end{aligned} \tag{7}$$

Plugging this back to Eq. (6), and for $D = O(\log n)$ (where we have roughly n^d different graphs with d vertices), we finally get

$$\|L_n^D\|^2 = \sum_{d=0}^{D} n^d \left(\frac{k}{n}\right)^{2d} = \sum_{d=0}^{D} \left(\frac{k^2}{n}\right)^d, \tag{8}$$

which remains bounded for $k > \sqrt{n}$ and blows up otherwise, predicting accurately the conjectured computational phase transition from the easy to the hard regime.

5. Statistical Query Lower Bounds for Detecting Planted Bicliques

Another approach to establishing the computational hardness of planted clique is by analyzing the detection variant where the statistical algorithm doing the detection is constrained to be a statistical query algorithm. Statistical query (SQ) algorithms are statistical algorithms that can only access data through statistical queries: that is, they cannot access individual data points, but can only compute statistics (expectations of real-valued functions) over a subsample of the data. The notion of SQ learning (Kearns, 1998) was originally introduced as a restriction of probably-approximately-correct (PAC) learning (Valiant, 1984) with more noise tolerance. One prominent development in cryptography based on SQs is the differential privacy framework for certifying that algorithms defined in terms of SQs cannot access individual records in a dataset (Dwork et al., 2014).

More relevant for our current purposes, Feldman et al. (2017) demonstrated the existence of a computational gap for a near variant of planted clique detection assuming the algorithm solving the problem may only interact with the data through statistical queries. We now introduce the problem they analyzed, the basic notions of statistical queries and statistical query dimension, and sketch their results.

5.1 Statistical Queries and Statistical Algorithms

This section recounts the formalism presented by Feldman et al. (2017). They define a statistical algorithm as an algorithm that interacts with some random data (given by a distribution $\mathcal{D}$) through SQs. A SQ is a call to a *STAT oracle*, which is a procedure that

estimates the expectation of a bounded real-valued function of the data. The calls to STAT may be adaptive, meaning that later calls may condition on previous ones. The STAT oracle can be defined as a high level of generality as follows:

Definition 3 (STAT oracle; Kearns, 1998) *Let $\mathcal{D}$ be a distribution over input domain $\mathcal{X}$. For $\tau > 0$, $\mathsf{STAT}(\tau)$ is the (randomized) oracle that, for any function $h : \mathcal{X} \to [-1, 1]$, returns some value v such that*

$$v \in [\mathbb{E}_{x\sim\mathcal{D}}[h(x)] - \tau, \mathbb{E}_{x\sim\mathcal{D}}[h(x)] + \tau] .$$

That is, $\mathsf{STAT}(\tau)$ oracle estimates the expectation of any function over the input, and the approximation error to the true value is guaranteed to be at most τ. This definition of STAT is a bit abstract; it may be easiest to understand it by thinking about how it might naturally be implemented. We could achieve a $\mathsf{STAT}(\tau)$ oracle by computing $h(x)$ on $O(1/\tau^2)$ samples from $\mathcal{D}$, and returning their empirical average. By standard learning-theoretic analysis, for all δ, the error of this empirical estimate is at most τ with probability $1 - \delta$, where the sample complexity is polynomial in both $1/\tau$ and $\log(1/\delta)$ (Kearns, 1998). Thus, any problem learnable in terms of SQs is also PAC-learnable (cf. Valiant, 1984).

The oracle that Feldman et al. (2017) use in their analysis is VSTAT, a modification of the standard STAT oracle where h is boolean-valued, and where tolerance is set a bit differently:

Definition 4 (VSTAT oracle; Feldman et al., 2017) *Let $\mathcal{D}$ be a distribution over input domain $\mathcal{X}$. For $t > 0$, $\mathsf{VSTAT}(t)$ is the (randomized) oracle that, for any function $h : \mathcal{X} \to \{0, 1\}$, returns some value $v \in [p - \tau, p + \tau]$, where $p = \mathbb{E}_{x\sim\mathcal{D}}[h(x)]$ and $\tau = \max\{1/t, \sqrt{\frac{p(1-p)}{t}}\}$.*

Note that $\mathsf{VSTAT}(t)$ can simulate $\mathsf{STAT}(1/\sqrt{t})$ and can be simulated by $\mathsf{STAT}(1/t)$ (Feldman et al., 2017), enabling conversion of results presented in VSTAT terms to STAT terms if desired.

5.2 Statistical Query Dimension

One measure of the complexity of a statistical algorithm defined in terms of SQs is its SQ dimension, which was originally proposed in the context of SQ learning problems (Blum et al., 1994), and later generalized to analyze arbitrary statistical algorithms (Feldman et al., 2017; Diakonikolas et al., 2022). We largely adapt the presentation from Feldman et al. (2017).

The precise definition of SQ dimension is quite technical, but intuitively it measures the average correlation between two distributions. To arrive at the notion of SQ dimension, we first define an inner product over functions $f, g : \mathcal{X} \to [-1, 1]$ with respect to a distribution (density or mass function) $\mathcal{D}$ as follows:

$$\langle f, g\rangle_{\mathcal{D}} = \mathbb{E}_{x\sim\mathcal{D}}[f(x)g(x)].$$

The norm of a function $f : \mathcal{X} \to [-1, 1]$ is thus

$$\|f\|_{\mathcal{D}} \triangleq \sqrt{\langle f, f\rangle_{\mathcal{D}}} = \sqrt{\mathbb{E}_{x\sim\mathcal{D}}[f(x)^2]}.$$

We can now use this norm to define the pairwise correlation $\chi_{\mathcal{D}}(\mathcal{D}_1, \mathcal{D}_2)$ between two distributions $\mathcal{D}_1, \mathcal{D}_2$ with respect to a reference distribution $\mathcal{D}$:

$$\chi_{\mathcal{D}}(\mathcal{D}_1, \mathcal{D}_2) = \left| \left\langle \frac{\mathcal{D}_1}{\mathcal{D}} - 1, \frac{\mathcal{D}_2}{\mathcal{D}} - 1 \right\rangle \right|.$$

In the special case where $\mathcal{D}_1 = \mathcal{D}_2$, this correlation recovers something familiar from classical statistics: the χ^2 distance between $\mathcal{D}_1$ and $\mathcal{D}$. The next step towards defining the SQ dimension is to generalize pairwise correlation between two distributions to the *average* correlation between pairs of distribution $\mathcal{D}_1, \mathcal{D}_2$ in some finite set of distributions $\mathbb{D}$ (still with respect to a reference distribution $\mathcal{D}$). This can be defined simply as the average pairwise correlation between all distributions in $\mathbb{D}$:

$$\begin{aligned} \rho(\mathbb{D}, \mathcal{D}) &\triangleq \frac{1}{|\mathbb{D}|^2} \sum_{\mathcal{D}_1, \mathcal{D}_2 \in \mathbb{D}} \chi_{\mathcal{D}}(\mathcal{D}_1, \mathcal{D}_2) \\ &= \frac{1}{|\mathbb{D}|^2} \sum_{\mathcal{D}_1, \mathcal{D}_2 \in \mathbb{D}} \left| \left\langle \frac{\mathcal{D}_1}{\mathcal{D}} - 1, \frac{\mathcal{D}_2}{\mathcal{D}} - 1 \right\rangle \right|. \end{aligned}$$

Consider a search problem Z (e.g., planted clique detection) over the domain $\mathcal{X}$ (e.g., graphs). Associated with Z is some set of distributions $\mathbb{D}$ over $\mathcal{X}$ and a class of solutions $\mathcal{F}$. We are finally in a position to define the SQ dimension of Z.

Definition 5 (SQ dimension; Feldman et al., 2017) *For $\bar{\gamma} > 0, \eta > 0$ and a search problem Z, we define the SQ dimension $\mathsf{SQD}(\mathsf{Z}, \bar{\gamma}, \eta)$ to be the largest integer d such that there exists a reference distribution $\mathcal{D}$ over $\mathcal{X}$ and a set $\mathbb{D}_{\mathcal{D}} \subseteq \mathbb{D}$ that satisfy the following:*

1. *For any solution $f \in \mathcal{F}$, the set $\mathbb{D}_f = \mathbb{D}_{\mathcal{D}} \setminus \mathsf{Z}^{-1}(f)$ has size at least $(1 - \eta) \cdot |\mathbb{D}_{\mathcal{D}}|$;*
2. *For any $f \in \mathcal{F}$ and $\mathbb{D}' \subseteq \mathbb{D}_f$ such that $|\mathbb{D}'| \geq |\mathbb{D}_f|/d$, it holds that $\rho(\mathbb{D}', \mathcal{D}) > \bar{\gamma}$.*

This definition allows us to prove the following main result, which describes the number of queries needed by a statistical algorithm to solve a problem as a function of the problem's SQ dimension:

Theorem 6 (Feldman et al., 2017) *Let Z be a search problem parameterized by distributions $\mathbb{D}$ over $\mathcal{X}$ and hypothesis class $\mathcal{F}$. For $\bar{\gamma} > 0$ and $\eta \in (0, 1)$, we let $d = \mathsf{SQD}(\mathsf{Z}, \bar{\gamma}, \eta)$. Then, any SQ algorithm that solves Z with probability $\delta > \eta$ requires at least $\frac{\delta - \eta}{1 - \eta}$ queries to the oracle $\mathsf{VSTAT}(1/3(\bar{\gamma})$.*

This will be the main tool enabling analysis of the bipartite planted (bi)clique problem.

5.3 SQ Analysis of Planted Biclique Detection

Bipartite planted (bi)clique is a variant of planted clique where the graph is bipartite, i.e., the nodes V can be partitioned into two sets A, B such that every undirected edge goes from A to B. In this kind of graph, we consider a set of nodes $N \subseteq V$ to be a *biclique* if all nodes in $N \cap A$ have an edge to all nodes in $N \cap B$. Going forward in this section, we will refer to a biclique simply as a clique.

(a) A random bipartite graph under the partition $\{\{1,2\},\{3,4\}\}$.

(b) Resulting graph after planting a biclique at nodes $\{1,3,4\}$.

Figure 3: Example of (a) a random bipartite graph and (b) the same graph with a planted biclique.

The generation of the random graph proceeds analogously to standard planted clique. For each possible edge between A and B, we choose to include it with probability $1/2$. We plant a clique by selecting a set N of k nodes and adding all edges between $N \cap A$ and $N \cap B$. Figure 3 shows a simple example of this process.

Analogously to standard planted clique (subsection 3.2), bipartite planted clique becomes statistically solvable when $k \geq (2+\epsilon)\log n$ (Feldman et al., 2017). Additionally, the algorithms from subsection 3.3 can be extended to solve it when $k \geq \Omega(\sqrt{n})$. However, unlike for standard planted clique, Feldman et al. (2017) are able to prove that computational hardness of solving this problem using an SQ algorithm when $k \leq n^{1/2-\delta}$ for $\delta > 0$. The idea is to first compute the SQ dimension of the bipartite planted clique problem (Theorem 7) and then apply Theorem 6 to get a lower bound on the query complexity and runtime (Theorem 8).

Theorem 7 (Feldman et al., 2017) *For all $\delta > 0$, let $k \leq n^{1/2-\delta}$, and let Z_k be bipartite planted k-clique problem. Then, for all $\ell \leq k$,*

$$\mathsf{SQD}\left(\mathsf{Z}_k, 2^{\ell+2}k^2/n^2, 1/\binom{n}{k}\right) \geq n^{2\ell\delta}/4.$$

By setting $\ell = \log r$ and applying Theorem 6, Feldman et al. (2017) obtain the following lower bound on the hardness of bipartite planted k-clique:

Theorem 8 (Feldman et al., 2017) *For any $\delta > 0$, $k \leq n^{1/2-\delta}$, and $r > 0$, any SQ algorithm must use at least $n^{\Omega(\log r)}$ queries to $\mathsf{VSTAT}(n^2/(rk^2))$ to solve bipartite planted k-clique. No poly-time SQ algorithm can solve the problem using queries to $\mathsf{VSTAT}(o(n^2/k^2))$ and any SQ algorithm will require $n^{\Omega(\log n)}$ queries to $\mathsf{VSTAT}(n^{2-\delta}/k^2)$.*

In summary, using SQ dimension, Feldman et al. (2017) were able to show a lower bound on the number of queries required to solve bipartite planted k-clique in the critical regime. Thus, they have demonstrated the existence of a statistical computational gap for this simplified version of the planted clique problem, at least under the simplifying assumption that the algorithm solving the problem is an SQ algorithm.

6. Conclusion

Many problems are thought to exhibit statistical-computational gaps, but proving that they do is challenging because it involves deriving super-polynomial lower bounds on any algorithm solving the problem. Planted clique is one of the longest studied of such problems, and thus perhaps one of the best candidates for establishing lower bounds to prove a gap. We have reviewed recent work that makes progress towards establishing such a gap assuming a restricted class of statistical inference algorithms (low-degree polynomial and statistical query algorithms). In addition to expanding our understanding of the planted clique problem, this line of research may help build a foundational understanding of what properties of a problem create a statistical-computational gap, hopefully allowing us to better predict when and why such gaps exist for more natural statistical problems in the real world.

References

Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262, 1994.

Zongchen Chen, Elchanan Mossel, and Ilias Zadik. Almost-linear planted cliques elude the metropolis process, 2022. URL `https://arxiv.org/abs/2204.01911`.

Ilias Diakonikolas, Daniel M Kane, and Yuxin Sun. Optimal sq lower bounds for robustly learning discrete product distributions and ising models. *arXiv preprint arXiv:2206.04589*, 2022.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh S Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *Journal of the ACM (JACM)*, 64(2):1–37, 2017. URL `https://www.cs.purdue.edu/homes/egrigore/papers/FGRVX13.pdf`.

David Gamarnik and Ilias Zadik. The landscape of the planted clique problem: Dense subgraphs and the overlap gap property, 2019. URL `https://arxiv.org/abs/1904.07174`.

Ilias Zadik. Computational and statistical challenges in high dimensional statistical models, 2019.

Mark Jerrum. Large cliques elude the metropolis process. *Random Structures & Algorithms*, 3(4):347–359, 1992. doi: https://doi.org/10.1002/rsa.3240030402. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.3240030402`.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Dmitriy Kunisky, Alexander S. Wein, and Afonso S. Bandeira. Notes on computational hardness of hypothesis testing: Predictions using the low-degree likelihood ratio. In Paula Cerejeiras and Michael Reissig, editors, *Mathematical Analysis, its Applications and Computation*, pages 1–50, Cham, 2022. Springer International Publishing.

Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021. doi: 10.1017/9781108637435.

Luca Trevisan. Lecture 1: Beyond worst-case analysis, 2017. URL `https://lucatrevisan.github.io/teaching/bwca17/lectures/lecture01.pdf`. Lecture notes from UC Berkeley.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Lenka Zdeborová and Florent Krzakala. Statistical physics of inference: thresholds and algorithms. *Advances in Physics*, 65(5):453–552, 2016.