Average-case complexity and statistical inference

We give an example of the sort of question we will look at. The idea is to 'hide' some structure within a high-dimensional random object. We may then ask if we can *detect* this, i.e. whether we can distinguish the vanilla random object from the random object plus planted structure, also if we may *recover*, i.e. 'find' the planted structure from within the random object.

A favourite combinatorial random object is the Erdős-Rènyi random graph, $G_{n,1/2}$, take n vertices and between each pair of vertices independently place an edge with probability $1/2$. We are interested in the typical behaviour for large n. The structure we plant is a clique, i.e. between a subset S^* of the vertices of the graph, we place all the possible edges. Now, the random graph $G_{n,1/2}$ without the planted structure will naturally have some cliques by chance, and indeed the largest of these will have size approximately $2\log_2 n$ with probability tending to 1 as $n \to \infty$; which suggests it might not be possible to detect or recover a planted structure of size smaller than $2\log_2 n$. This turns out to be true, as we shall see. Interestingly, there is another phase transition. Fast algorithms finding the clique, e.g. picking the vertices of highest degrees, are only known when the planted clique has size about $n^{1/2}$ or higher; which is considerably larger than $2\log_2 n$. There is some 'evidence' that this $n^{1/2}$ threshold is fundamental: by evidence we mean rigorous statements we can prove which *suggest* that there are no polynomial time algorithms.

These ideas will be made precise in the course as we investigate what forms this evidence can take. We illustrate the techniques on three running examples, planted clique, as described above, as well a generalisation of it planted dense subgraph, and a Gaussian planted structure problem biclustering - see Appendix A for a list and the phase transitions in each model. This area is very active and many of the techniques and results presented here have been developed within the last decade and some within the last year.

1 Lecture 1: Intro

1.1 Planted problems and what it means to succeed

2 Planted clique $PC(n,k)$

Writing $k = n^\alpha$

1. for $\alpha > n^{1/2+\varepsilon}$ EASY
   (a) sum test.
   (b) degree?
2. for $(2+\varepsilon)\log n < \alpha < n^{1/2-\varepsilon}$ conjectured HARD
   (a) low degree algorithms fail
   (b) brute force search statistic succeeds
3. for $\alpha < (2-\varepsilon)\log n$ IMPOSSIBLE

3 Bi-clustering $BC(n,k,\lambda)$

4 Planted dense subgraph $PDS(n,k,p,q)$

Writing $k = n^\alpha$ and $\lambda = n^{-\beta}$ we have the following phases

1. for α, β EASY we have efficient algorithms which distinguish with probability tending to 1 as $n \to \infty$

   (a) ?

2. for α, β conjectured HARD we have brute-force algorithms which distinguish with probability tending to 1 as $n \to \infty$ and evidence of hardness

   (a) reduction (see p. ?)

3. for α, β conjectured HARD we have brute-force algorithms which distinguish with probability tending to 1 as $n \to \infty$ and evidence of hardness

   (a) reduction (see p. ?)

This is how it is $q_{\mathcal{A}}^{D}(G)$
$PC(n, k)$
This course will introduce some theory of, and cover some recent results in computational complexity of statistical inference problems - detecting/recovering signals in noisy data.

First lecture: Thursday 26th Jan 15:15-17.00 in room 64119.

Details: The course is 5hp and will run throughout the spring term to June. Assessment will take the form of 2 exercise sheets and 1 longer piece to focus on tractability/hardness/impossibility of a particular problem not covered in the lectures or possibly a novel application of our methods assessed via a written report or talk.

Lecturer: Fiona Skerman, `fiona.skerman@math.uu.se`. Feel free to contact me with any questions about the course.

Description:

Statistical inference to us is detecting or recovering a signal in noisy data. An example is finding a submatrix with entries normally distributed with mean λ and variance one in a matrix with all other entries normally distributed with mean zero and variance one. For some signal strengths λ and some sizes of submatrix fast algorithms are known which succeed with probability near 1; for some regions of parameters it is information theoretically impossible to succeed with probability near 1.

We are interested also in the third region - parameter values for which there are brute-force algorithms which succeed but no fast algorithms are known. In particular there is a regime of parameter values where finding the submatrix is conjectured hard: and we are interested in rigorous results which give evidence of hardness i.e. showing failure of restricted classes of algorithms and showing average-case reductions to problems we believe are hard.

The course will use two running example problems, finding / detecting a submatrix with elevated mean in a large random matrix, and finding / detecting a dense subgraph within a large random graph; and will establish the phase transition diagrams for these problems (see below). These exhibit a detection-recovery gap, it is easier to detect the presence of the planted substructure than to recover it even approximately. We will also cover definitions, ideas and techniques necessary to establish phase-transitions of hardness for statistical inference problems: including analysis of algorithms on

random structures: spectral techniques, SDPs & brute-force, bounding chi-square divergence, low-degree polynomial method, average-case reductions. Note the probabilistic aspect means one has to be careful what a reduction is (it is allowed to fail on some instances for example) and the proofs have different techniques.

The following phase transition diagrams show parameter regimes where the example problems are **easy** (fast algorithms succeed with probability near 1), **hard** (brute-force/slow algorithms succeed with probability near 1 and evidence no such fast algorithms exits) and **impossible** (information theoretically impossible for any algorithm to succeed with probability near 1). In each diagram the size $\sim k(n)$ of the planted structure increases along the y-axis and the strength of the signal of the planted substructure decreases along the x-axis. The **detection problem** is, given a (random) sample from either H_0, a distribution with no planted substructure or H_1, a distribution with a planted substructure to determine which distribution it likely came from. The **recovery problem** is, given a (random) sample from H_1 a distribution with a planted substructure, output (exactly or approximately) the location of the planted substructure.

A Planted problems

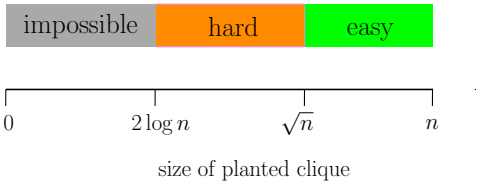
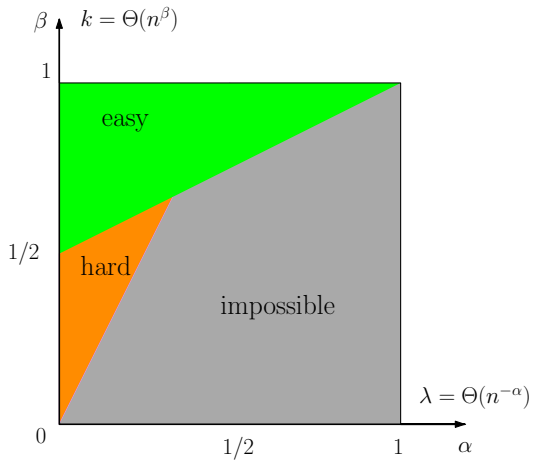
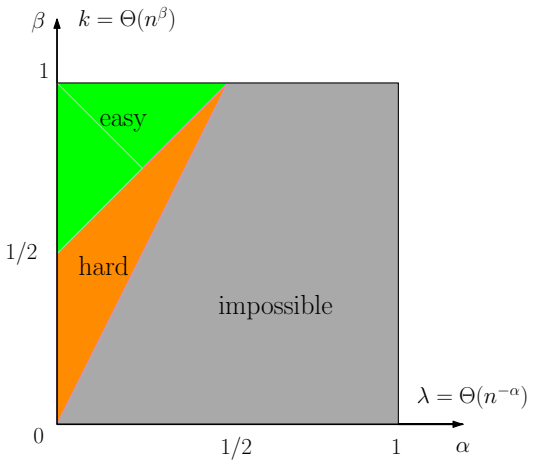


Figure 0: **Planted clique.**
H_0: $G(n, \frac{1}{2})$ random graph on n vertices where each edge is present independently with probability 1/2.
H_1: $G(n, k, \frac{1}{2})$, random graph on n vertices where each vertex is part of 'community' S independently with probability k/n. Each edge ij is present independently either with probability 1 if $i, j \in S$ or with probability 1/2 otherwise.

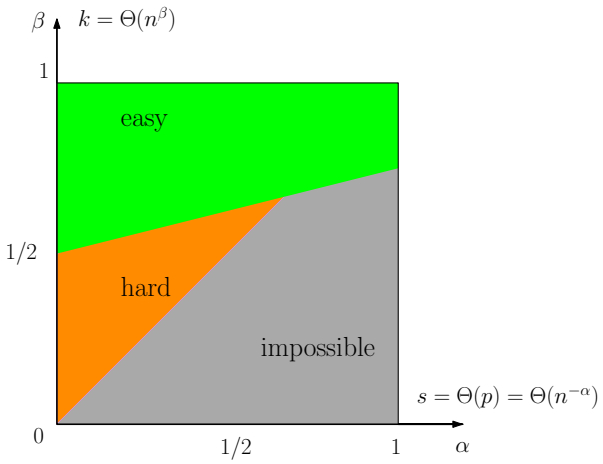


(a) detection

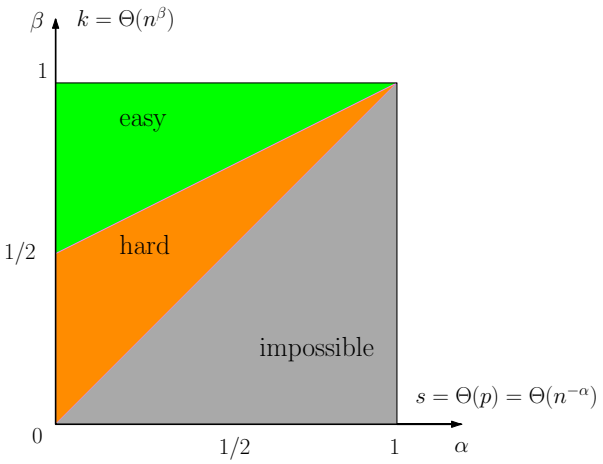


(b) recovery

Figure 1: **Spiked Matrix Model** (planted submatrix with elevated mean).
H_0: random $n \times n$ matrix with each entry independent with distribution $N(0, 1)$.
H_1: $n \times n$ matrix with each index in set S independently with probability k/n. Each entry independent with distribution $N(\lambda, 1)$ if $i, j \in S$ and with distribution $N(0, 1)$ otherwise.



(a) detection



(b) recovery

Figure 2: **Planted dense subgraph**.
H_0: $G(n, q)$ random graph on n vertices where each edge is present independently with probability q.
H_1: $G(n, k, q, s)$ with $s > 0$, random graph on n vertices where each vertex is part of 'community' S independently with probability k/n. Each edge ij is present independently either with probability $q + s$ if $i, j \in S$ or with probability q otherwise.

Index