A Theorist's Toolkit: Property testing

Notes taken by Jakob Nordström

March 10, 2005

This lecture was given by Douglas Wikström based on the paper *Property Testing and Its Connection to Learning and Approximation* by Goldreich, Goldwasser and Ron (Journal of the ACM, vol. 45, no. 4, July 1998, pp. 653–750).

1 Introduction

Loosely speaking, the goal of property testing is to determine whether a certain object has a certain property. Let us represent our objects as functions f. Then a property P can be represented as the set of functions $\mathcal{F}$ having the property in question.

For example, we might encode a graph G as the indicator function f for the edges in G. The property of, say, bipartiteness of a graph is then represented naturally as the set of all f encoding bipartite graphs.

The first naive goal can be expressed as follows.

Goal 1 (Decision task). *Given a set of functions $\mathcal{F}$ and a function f, determine efficiently whether $f \in \mathcal{F}$ or not.*

This turns out to be too hard. Suppose for $f : \{0,1\}^n \mapsto \{0,1\}$ that there is a $g \in \mathcal{F}$ such that f and g differ in only one point. There is no reasonable way to find this point efficiently (where by "efficiently" we mean in time $\mathrm{poly}(n)$, for example). Instead we try to reach a less ambitious goal.

Goal 2 (Property testing (almost)). *Given a set of functions $\mathcal{F}$ and a function f, determine efficiently and with low probability of error whether f is close to some function in $\mathcal{F}$ or not.*

This is closer to, but still not quite, what we need. We have to determine what to do with the functions which are on the borderline between being "close" and "far away". If this border is sharp, the same example as above shows that there will be cases on the border that are impossible to distinguish efficiently and with high probability. Thus if we want efficient tests, we need to accept a grey area in between "$f \in \mathcal{F}$" and "f far from $\mathcal{F}$" where the test results may be inconclusive.

We are now ready to stop beating around the bush and get down to business. The rest of these notes is organized as follows. In section 2 we give the formal definitions, and section 3 collects some facts that we use in the proofs. In section 4 we show that some properties are very hard to test, and this is followed by some general observations about testability of unions, intersections and complements of testable properties in section 5. Finally, in section 6 we present a property tester for graph bipartiteness.

Property testing have natural connections with learning theory and approximation. We do not pursue these connections in these notes, however, but refer the interested reader to the (very comprehensive) article by Goldreich, Goldwasser and Ron.

2 Formal Definitions

The general goal of property testing as stated in the article by Goldreich, Goldwasser and Ron is the following.

Goal 3 (Property testing). *Let $\mathcal{F}$ be a set of functions encoding a fix property P, and let f be an unknown function. Our goal is to determine (possibly probabilistically) if $f \in \mathcal{F}$ or if it is far from all functions in $\mathcal{F}$, where distance between functions is measured with respect to some distribution D on the domain of f. Towards this end, we are given examples of the form $(x, f(x))$, where x is distributed according to D. We may also be allowed to query f on instances of our choice.*

Note that we do not specify what should happen if $f \notin \mathcal{F}$ but f is close to some function in $\mathcal{F}$. Thus we do not require that functions close to $\mathcal{F}$ should be recognized.

Returning to the example in section 1, for bipartite graphs it is clear that one way of "testing" bipartiteness is to solve the corresponding decision task, and that this task can be solved very efficiently. When talking about property testing we are therefore interested only in algorithms that yield a dramatic speed-up compared to the algorithms for the corresponding decision task.

The two most relevant parameters to property testing are the permitted distance ϵ and the desired confidence δ. We require the tester to accept each function in $\mathcal{F}$ and reject every function that is further than ϵ away from any function in $\mathcal{F}$. The tester can be probabilistic, and may make incorrect positive and negative assertions with probability at most δ. The interesting complexity measures are the *sample complexity* (the number of examples function values that the tester requires), the *query complexity* (the number of function queries made, if we allow this), and the *running time* of the property tester.

We let $\mathcal{F} = \left\{\mathcal{F}_n\right\}_{n=1}^{\infty}$ denote a family of sets of functions, where $\mathcal{F}_n$ contains functions from $\{0,1\}^n$ to $\{0,1\}$, i.e. $\mathcal{F}_n \subseteq \left\{\{0,1\}^n \mapsto \{0,1\}\right\}$. Also, $\mathcal{D}$ denotes a family of probability distributions $\left\{\mathcal{D}_n\right\}_{n=1}^{\infty}$, where $\mathcal{D}_n$ is a distribution on $\{0,1\}^n$.

We consider two functions close if they are likely to coincide on random input drawn from $\mathcal{D}_n$. Let $x \sim \mathcal{D}_n$ denote that x is a random sample from $\mathcal{D}_n$.

Definition 2.1 (ϵ-close). The functions $f, g : \{0,1\}^n \mapsto \{0,1\}$ are ϵ-*close* with respect to $\mathcal{D}_n$ if

$$\Pr_{x \sim \mathcal{D}_n}[f(x) \neq g(x)] \leq \epsilon$$

and ϵ-*far* otherwise. For a set of functions $\mathcal{F}_n$, f is ϵ-close to $\mathcal{F}_n$ with respect to $\mathcal{D}_n$ if there exists a $g \in \mathcal{F}_n$ such that f and g are ϵ-close and ϵ-*far* otherwise.

The basic definition of a property tester that we will use is as follows.

Definition 2.2 (Property tester). For a function f and any probability distribution $\mathcal{D}_n$, let $E = \big[\big((x_1, f(x_1)\big), \big((x_2, f(x_2)\big), \ldots\big]$ be a set of *f-labelled examples*, where each x_i is independently drawn from the distribution $\mathcal{D}_n$.

The algorithm $A = A(n, \epsilon, \delta, f, E)$ is a *property tester* for $\mathcal{F}$ if for all n, f, $\mathcal{D}_n$, $\epsilon \in \big(0,1\big)$ and $\delta \in \big(0, \frac{1}{2}\big)$ it holds that

$$\begin{aligned} &\Pr\left[A(n,\epsilon,\delta,f,E) = 1\right] \geq 1 - \delta \qquad && \text{if } f \in \mathcal{F}_n \text{ and} \\ &\Pr\left[A(n,\epsilon,\delta,f,E) = 1\right] \leq \delta && \text{if } f \text{ is } \epsilon\text{-far from } \mathcal{F}_n \text{ w.r.t. } \mathcal{D}_n, \end{aligned}$$

where the probability is over the example inputs drawn from $\mathcal{D}_n$ and the possible coins tosses of A.

The *sample complexity* of A is a function of n, ϵ and δ bounding the number of labeled examples examined by A on input (n, ϵ, δ). The *time complexity* of A is a function of n, ϵ and δ bounding the running time of A on input (n, ϵ, δ).

Note in particular that A should work for any probability distribution $\mathcal{D}_n$, however strangely biased it may be. This is not as unreasonable as it might seem, however, since A most likely will be tested on the input that $\mathcal{D}_n$ "considers important".

There are a number of flavours of definition 2.2. For instance:

- $\mathcal{D}_n$ might be a known fixed distribution (in particular, we will be interested in testing with respect to the uniform distribution) or might be restricted to some known class of distributions.

- The algorithm may be given oracle access to f, meaning that it can ask questions for $x \in \{0,1\}^n$ and receive $f(x)$ as answers. In this case, we refer to the number of queries made by A as a function of n, ϵ and δ as the *query complexity* of A.

- We can demand a certificate for the fact that $f \notin \mathcal{F}_n$ if A rejects f. Certificates are defined with respect to a verification algorithm that accepts a sequence of labelled examples whenever there exists a $f \in \mathcal{F}_n$ which is consistent with the sequence. We do not require that the algorithm should reject each sequence which is not consistent with any $f \in \mathcal{F}_n$. A certificate for $f \notin \mathcal{F}_n$ is then an f-labelled sequence which is rejected by the verification algorithm.

- The algorithm may have only one-sided error, i.e. it always accepts if $f \in \mathcal{F}$.

- Two distance parameters ϵ_1, ϵ_2 can be given, where we require A to accept with high probability every f which is ϵ_1-close to $\mathcal{F}_n$ and reject with high probability every f which is ϵ_2-far from $\mathcal{F}_n$.

3 Some Useful Stuff

In this section we have collected some more or less well-known facts that will be of help during the rest of this lecture. We start off by stating Markov's inequality for reference.

Fact 3.1. *Let X be a non-negative random variable with finite mean $\mathrm{E}\big[X\big]$. Then for all $K \in \mathbb{R}^+$ it holds that*

$$\Pr\left[X \geq K\right] \leq \frac{\mathrm{E}\big[X\big]}{K}$$

Next follow some Chernoff bounds.

Fact 3.2. *Let $X_1, \ldots, X_m$ be independent and identically distributed random variables with $\Pr\left[X_i = 1\right] = \Pr\left[X_i = -1\right] = \frac{1}{2}$. Then*

$$\Pr\left[\sum_{i=1}^{m} X_i \geq a\right] \leq \exp\left(-\frac{a^2}{2m}\right)$$

Noting that $\Pr\left[\frac{X_i+1}{2} = 1\right] = \Pr\left[\frac{X_i+1}{2} = 0\right] = \frac{1}{2}$ and using symmetry, we get the following corollary.

Corollary 3.3. *Let $Y_1, \ldots, Y_m$ be independent and identically distributed random variables with $\Pr\left[Y_i = 1\right] = \Pr\left[Y_i = 0\right] = \frac{1}{2}$ and suppose that $c < \frac{1}{2}$. Then*

$$\Pr\left[\sum_{i=1}^{m} Y_i \leq cm\right] \leq \exp\left(-\frac{(2c-1)^2}{2}m\right).$$

Fact 3.4. *Let $X_1, \ldots, X_m$ be independent (but not necessarily identically distributed) random variables with $X_i \in \big[0, 1\big]$, and let $X = \frac{1}{m}\sum_{i=1}^{m} X_i$. Then for $\gamma \in \big[0, 1\big]$ it holds that*

$$\Pr\left[X > (1+\gamma)\mathrm{E}\left[X\right]\right] < \exp\left(-\frac{1}{3}\gamma^2 m\mathrm{E}\left[X\right]\right),$$
$$\Pr\left[X < (1-\gamma)\mathrm{E}\left[X\right]\right] < \exp\left(-\frac{1}{2}\gamma^2 m\mathrm{E}\left[X\right]\right).$$

We mention two inequalities used for estimating binomal coefficients which may come in handy.

Fact 3.5. *For $n > m > k > 1$ it holds that*

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$$

and

$$\frac{\binom{m}{k}}{\binom{n}{k}} < \left(\frac{m}{n}\right)^k.$$

We will be interested in studying the statistical difference of random variables. In particular, we will need the result that no algorithm can distinguish between samples from two random variables with higher accuracy than their statistical difference.

Definition 3.6. Let X and Y be two random variables on $\{0,1\}^n$. The *statistical difference* of X and Y is

$$\Delta\big(X, Y\big) = \frac{1}{2}\sum_{\alpha \in \{0,1\}^n} \Big|\Pr\left[X = \alpha\right] - \Pr\left[Y = \alpha\right]\Big|.$$

The following definition is probably slightly non-standard. Usually it is required that the algorithm A runs in polynomial time.

Definition 3.7. Let X_1 and X_2 be two random variables on $\{0,1\}^n$. Let $A = A : \{0,1\}^n \mapsto \{0,1\}$ be a probabilistic algorithm and let

$$E_A(X_i) = \sum_{\alpha \in \{0,1\}^n} \Pr[X_i = \alpha] \cdot \Pr[A(\alpha) = 1]$$

denote the probability that A outputs 1 on a sample from X_i. We note that the second probability is over the internal random choices of A conditioned on the specific value of α. Then A is a γ-*distinguisher* for X_1 and X_2 if

$$|E_A(X_1) - E_A(X_2)| \geq \gamma.$$

It is intuitively clear that no distinguisher can beat the statistical difference.

Proposition 3.8. *Suppose that X and Y are random variables on $\{0,1\}^n$ with $\Delta\big(X, Y\big) = \delta$. Then there can be no γ-distinguisher for X and Y with $\gamma > \delta$.*

Proof. Using that

$$\begin{aligned} &\sum_{\alpha \in \{0,1\}^n} \Pr[A(\alpha) = 0] \cdot \big(\Pr[X = \alpha] - \Pr[Y = \alpha]\big) \\ = -&\sum_{\alpha \in \{0,1\}^n} \Pr[A(\alpha) = 1] \cdot \big(\Pr[X = \alpha] - \Pr[Y = \alpha]\big), \end{aligned}$$

which is true since $\sum_\alpha \Pr[X = \alpha] - \Pr[Y = \alpha] = 0$, we get for any A that

$$\begin{aligned} |E_A(X) - E_A(Y)| &= \left|\sum_\alpha \Pr[A(\alpha) = 1] \cdot \big(\Pr[X = \alpha] - \big(\Pr[Y = \alpha]\big)\right| \\ &= \frac{1}{2} \sum_{i \in \{0,1\}} \left|\sum_\alpha \Pr[A(\alpha) = i] \cdot \big(\Pr[X = \alpha] - \big(\Pr[Y = \alpha]\big)\right| \\ &\leq \frac{1}{2} \sum_{i \in \{0,1\}} \sum_\alpha \Pr[A(\alpha) = i] \cdot \left|\big(\Pr[X = \alpha] - \big(\Pr[Y = \alpha]\big)\right| \\ &= \frac{1}{2} \sum_\alpha \left|\big(\Pr[X = \alpha] - \big(\Pr[Y = \alpha]\big)\right| \\ &= \Delta\big(X, Y\big) \end{aligned}$$

and the proposition follows. □

4 The Bad News: Some Properties Are Hard

The bad news is that not all properties are efficiently testable. The counterexample below is highly nonconstructive, however.

Theorem 4.1. *There is a family of sets of functions $\mathcal{F} = \big\{\mathcal{F}_n\big\}_{n=1}^{\infty}$ for $\mathcal{F}_n \subseteq \big\{\{0,1\}^n \mapsto \{0,1\}\big\}$ such that every property tester A must look at a constant* *fraction of the domain (i.e. on $\Omega(2^n)$ inputs). This holds even for testing with respect to the uniform distributions, for any constant distance parameter $\epsilon < 1/2$ and confidence parameter $\delta < 1/2$, and even when allowing the algorithm to make queries and use unlimited computing time.*

Proof. We use the probabilistic method to demonstrate the existence of a function class $\mathcal{F} = \{\mathcal{F}_n\}_{n=1}^{\infty}$, where $|\mathcal{F}_n| = 2^{\Omega(2^n)}$, satisfying the statement of the theorem. The outline of the proof is as follows.

First we show that a random function $g : \{0,1\}^n \mapsto \{0,1\}$ is very likely to be far from $\mathcal{F}_n$ no matter how $\mathcal{F}_n$ is chosen. Note that if g is selected uniformly at random, a sequence of T example function values of g is just T uniformly distributed random bits. We then study the statistical difference between these random bits and the distribution of function values induced by a uniformly selected function in $\mathcal{F}_n$. The idea is that if the number of example input-output pairs T is sufficiently small, the example functions values of $f \in \mathcal{F}_n$ and a random ϵ-far function g will be sufficiently close statistically so that a property tester A can impossibly distinguish with particularly high accuracy whether the sample is from f or g. But this is exactly what A is supposed to do! By definition $f \in \mathcal{F}_n$ should be accepted with probability at least $1 - \delta$ while g must not be accepted with probability higher than δ. Contradiction.

Now for the details. Let $N = 2^n$. We do the case for $\epsilon = \frac{1}{4}$, $\delta = \frac{1}{5}$ and a probabilistically constructed $\mathcal{F}_n$ of size $|\mathcal{F}_n| = 2^{\frac{1}{10}2^n}$, leaving it to the worried reader to generalize the proof to arbitrary ϵ and δ by modifying the size of $\mathcal{F}_n$ accordingly.

We claim that for a fix $\mathcal{F}_n$, a uniformly random $g : \{0,1\}^n \mapsto \{0,1\}$ is almost surely ϵ-far from $\mathcal{F}_n$. Since g is random, for a fixed f and all $y \in \{0,1\}^n$ it holds that $X_y = f(y) \oplus g(y)$ are independent random variables with $\Pr[X_y = 1] = \Pr[X_y = 0] = \frac{1}{2}$. We get

$$\begin{aligned}
\Pr_g[g \text{ is } \epsilon\text{-close to } \mathcal{F}_n] &\le \Pr_g\left[\exists f \in \mathcal{F}_n \text{ s.t. } \left|\{x : f(x) \neq g(x)\}\right| \le \epsilon N\right] \\
&\le |\mathcal{F}_n| \cdot \Pr_g\left[\left|\{x : f(x) \oplus g(x) \neq 0\}\right| \le \epsilon N\right] \\
&\le 2^{N/10} \cdot \Pr\left[\sum_{i=1}^{N} X_i \le \epsilon N\right] \\
&\le 2^{N/10} \cdot \exp(-N/8) \le \exp(-N/40), \qquad (1)
\end{aligned}$$

using the Chernoff bound in corollary 3.3 with $c = \epsilon = 1/4$ to get the next-to-last inequality. This proves the claim.

We now want to show that there exists a set of functions $\mathcal{F}_n$ such that that even a large sample of example input-output pairs from a random g is hard to distinguish from that of some $f \in \mathcal{F}_n$. We do this by picking $\mathcal{F}_n$ at random and arguing that it has the desirable property with overwhelming probability.

Fix a sequence $S = x_1, x_2, \ldots, x_T$ of $T = N/20$ inputs. Let us use vector notation $f(S)$ to mean $f(S) = f(x_1), f(x_2), \ldots, f(x_T)$. We want to compare the values $g(S)$ and $f(S)$ for a random g and some uniformly chosen function $f \in \mathcal{F}_n$. As was noted above, $g(S)$ is the uniform distribution. Let $\delta_S(\mathcal{F}_n)$ denote the statistical difference between the uniform distribution and the distribution of function values induced by a uniformly selected function in $\mathcal{F}_n$, that is

$$\delta_S\big(\mathcal{F}_n\big) = \frac{1}{2} \sum_{\alpha \in \{0,1\}^T} \left| \Pr_{f \sim \mathcal{F}_n} [f(S) = \alpha] - 2^{-T} \right|. \tag{2}$$

The probability over a random choice of $\mathcal{F}_n$ that $\delta_S\big(\mathcal{F}_n\big) > 1/2$ is

$$\begin{aligned}
&\Pr_{\mathcal{F}_n}\left[\delta_S\big(\mathcal{F}_n\big) > \frac{1}{2}\right] \\
&\quad \le \Pr_{\mathcal{F}_n}\left[\exists \alpha \in \{0,1\}^T \text{ s.t. } \left| \Pr_{f \sim \mathcal{F}_n} [f(S) = \alpha] - 2^{-T} \right| > 2^{-(T+1)}\right] \\
&\quad \le 2^T \cdot \Pr_{\mathcal{F}_n}\left[\left| \Pr_{f \sim \mathcal{F}_n} [f(S) = \alpha] - 2^{-T} \right| > 2^{-(T+1)}\right].
\end{aligned} \tag{3}$$

Instead of just hitting hard with the right Chernoff bound and conclude the computation elegantly in one final line (as is done in the article), we pause to introduce some notation which hopefully will make it clearer what is happening. For a fixed α, let G_α be the random variable

$$G_\alpha = G_\alpha(\mathcal{F}_n) = \Pr_{f \sim \mathcal{F}_n} [f(S) = \alpha].$$

What we want to estimate in (3) is then

$$\Pr_{\mathcal{F}_n}\left[\big|G_\alpha(\mathcal{F}_n) - 2^{-T}\big| > 2^{-(T+1)}\right].$$

Let

$$X_{f,\alpha} = X_{f,\alpha}(\mathcal{F}_n) = \begin{cases} 1 & \text{if } f(S) = \alpha, \\ 0 & \text{otherwise,} \end{cases}$$

be the indicator function for whether $f(S) = \alpha$ or not. Then we can write G_α in terms of $X_{f,\alpha}$ as

$$G_\alpha = \frac{1}{\big|\mathcal{F}_n\big|} \sum_{f \in \mathcal{F}_n} X_{f,\alpha} \tag{4}$$

(simply the number of "good" choices divided by the total number of choices) and we get that

$$\Pr_{\mathcal{F}_n}\left[\big|G_\alpha - 2^{-T}\big| > 2^{-(T+1)}\right] = \Pr_{\mathcal{F}_n}\left[\left| \sum_{f \in \mathcal{F}_n} \frac{X_{f,\alpha}}{\big|\mathcal{F}_n\big|} - 2^{-T} \right| > 2^{-(T+1)}\right]. \tag{5}$$

Note that $\mathrm{E}\big[X_{f,\alpha}/\big|\mathcal{F}_n\big|\big] = 2^{-|\mathcal{F}_n|} \cdot 2^{-T}$, so what we are asking is what the probability is that a sum of $\big|\mathcal{F}_n\big|$ such random variables diverges from its expected value $\mu = 2^{-T}$ by more than $\mu/2$. Assuming that the $X_{f,\alpha}$ are independent (*which in fact is not quite true; see remark 4.2*) and using the Chernoff bounds in fact 3.4 with $\gamma = 1/2$, we can bound this probability by

$$\Pr_{\mathcal{F}_n}\left[\left| \sum_{f \in \mathcal{F}_n} \frac{X_{f,\alpha}}{\big|\mathcal{F}_n\big|} - 2^{-T} \right| > 2^{-(T+1)}\right] \le 2 \exp\left(-\frac{1}{3} \cdot \left(\frac{1}{2}\right)^2 \cdot 2^{-T}\big|\mathcal{F}_n\big|\right) \tag{6}$$

and putting together (3), (5) and (6) and recalling that $T = 2^{N/20}$ and $\left|\mathcal{F}_n\right| = 2^{N/10}$ we see that

$$\Pr_{\mathcal{F}_n}\left[\delta_S\big(\mathcal{F}_n\big) > \frac{1}{2}\right] \leq 2^{N/20} \cdot 2\exp\left(-\frac{1}{12} \cdot 2^{-N/20} \cdot 2^{N/10}\right) = \exp\left(-2^{\Omega(N)}\right). \quad (7)$$

Summing (7) over all $\binom{N}{T} < 2^N$ choices of S, we get that

$$\Pr_{\mathcal{F}_n}\left[\forall S \;\; \delta_S\big(\mathcal{F}_n\big) \leq \frac{1}{2}\right] \geq 1 - 2^N \exp\left(-2^{\Omega(N)}\right) \geq 1 - \exp\left(-2^{\Omega(N)}\right), \quad (8)$$

i.e. the probability that a random $\mathcal{F}_n$ is such that all $f(S)$ have distance at most $1/2$ from the uniform random distribution is overwhelming.

Fix such an $\mathcal{F}_n$. Using definition 3.7 and proposition 3.8, we see that a property tester A looking at $N/20$ inputs must be a good distinguisher of $f(S)$ for $f \in \mathcal{F}_n$ from uniformly random bits $g(S)$. In fact, a distinguisher better than the statistical difference $1/2$, as it turns out! More precisely, if we let r_A denote the coin tosses of A, by definition 2.2 we have

$$\Pr_{g\ \epsilon\text{-far}, r_A}\left[A(n,\epsilon,\delta,g) = 1\right] \leq \max_{g\ \epsilon\text{-far}}\left\{\Pr_{r_A}\left[A(n,\epsilon,\delta,g) = 1\right]\right\} \leq \delta \quad (9)$$

and

$$\Pr_{f\in\mathcal{F}_n, r_A}\left[A(n,\epsilon,\delta,f) = 1\right] \geq \min_{f\in\mathcal{F}_n}\left\{\Pr_{r_A}\left[A(n,\epsilon,\delta,f) = 1\right]\right\} \geq 1-\delta \quad (10)$$

but since $\mathcal{F}_n$ has distance most $1/2$ from the uniform random distribution this yields

$$\begin{aligned}
\frac{1}{2} &\geq \left|\Pr_{f\in\mathcal{F}_n, r_A}\left[A(n,\epsilon,\delta,f) = 1\right] - \Pr_{g, r_A}\left[A(n,\epsilon,\delta,g) = 1\right]\right| \\
&= \Big|\Pr\left[A(n,\epsilon,\delta,f) = 1\right] \;\; - \Pr\left[A(n,\epsilon,\delta,g) = 1 \mid g\ \epsilon\text{-far}\right] \cdot \Pr\left[g\ \epsilon\text{-far}\right] \\
&\qquad\qquad - \Pr\left[A(n,\epsilon,\delta,g) = 1 \mid g\ \epsilon\text{-close}\right] \cdot \Pr\left[g\ \epsilon\text{-close}\right]\Big| \\
&\geq (1-\delta) - \delta \cdot \big(1 - \exp\big(-\Omega(N)\big)\big) - 1 \cdot \exp\big(-\Omega(N)\big) \\
&\geq \frac{3}{5} - \exp\big(-\Omega(N)\big) \qquad (11)
\end{aligned}$$

which is a contradiction. ☐

Note that the complexity $\Omega\big(2^n\big)$ holds even if we assume that the tester is given oracle access to the tested functions and makes queries of its own, rather than just receiving a sample of examples.

Remark 4.2. The variables $X_{f,\alpha}\big(\mathcal{F}_n\big)$ for $f \in \mathcal{F}_n$ are not independent (though they seem to be identically distributed, although we do not use this), so formally speaking the Chernoff bounds in fact 3.4 are not applicable. This detail can be taken care of quite easily by picking $\mathcal{F}_n$ as a multiset (i.e. with replacement) and then show that

1. with very high probability all functions in $\mathcal{F}_n$ are distinct, and
2. if they are not and $\mathcal{F}_n$ is in fact a multiset, the proof still goes through.

5 The Algebra of Property Testing

Due to lack of time the material in this section was omitted from the lecture. Since it is a quite nice (and easy) application of the result in theorem 4.1, we include it in the lecture notes.

Given that we know that two properties P' and P'' are testable (within certain complexities), what can we say about the properties $P' \vee P''$, $P' \wedge P''$ and $\neg P'$? Or, speaking in terms of families of sets of functions $\mathcal{F}' = \{\mathcal{F}'_n\}$ and $\mathcal{F}'' = \{\mathcal{F}''_n\}$, what can be said about $\{\mathcal{F}'_n \cup \mathcal{F}''_n\}$, $\{\mathcal{F}'_n \cap \mathcal{F}''_n\}$ and $\{\overline{\mathcal{F}'_n}\}$?

Theorem 5.1. *Suppose that $\mathcal{F}' = \{\mathcal{F}'_n\}$ and $\mathcal{F}'' = \{\mathcal{F}''_n\}$, are testable with sample complexity $c'(n, \epsilon, \delta)$ and $c''(n, \epsilon, \delta)$, respectively. Then $\{\mathcal{F}'_n \cup \mathcal{F}''_n\}$ is testable with sample complexity $c'(n, \epsilon, \delta/2) + c''(n, \epsilon, \delta/2)$.*

Proof. Run the tester A' for $\mathcal{F}'$ and then the tester A'' for $\mathcal{F}''$, and accept if at least one of A' and A'' accepts. If $f \in \mathcal{F}'_n \cup \mathcal{F}''_n$, clearly the acceptance probability is at least $1 - \delta/2$. If f is ϵ-far from $\mathcal{F}'_n \cup \mathcal{F}''_n$ it is ϵ-far from both $\mathcal{F}'_n$ and $\mathcal{F}''_n$, and by the union bound the probability of erroneous acceptance is at most $\delta/2 + \delta/2 = \delta$. □

The key observation in the proof of theorem 5.1 is that the fact that f is ϵ-far from $\mathcal{F}'_n \cup \mathcal{F}''_n$ implies that it is ϵ-far from both $\mathcal{F}'_n$ and $\mathcal{F}''_n$. This is not true for intersection, which is why it does not hold in general that an intersection of testable properties are testable.

Theorem 5.2. *There exist families of sets of functions $\{\mathcal{F}'_n\}$ and $\{\mathcal{F}''_n\}$ which are trivially testable under the uniform distribution (i.e. by an oblivious tester that always accepts provided that $\epsilon > 2^{-n}$) but for which $\{\mathcal{F}'_n \cap \mathcal{F}''_n\}$ is not testable under the uniform distribution with query complexity $o(2^n)$ even for constant $\epsilon, \delta < 1/2$.*

Proof. For $i = 0, 1$, define $\mathcal{F}^i_n = \{f : \{0,1\}^n \mapsto \{0,1\} : f(0^n) = i\}$. Note that $\mathcal{F}^0_n$ and $\mathcal{F}^1_n$ are both trivially testable since every function $f : \{0,1\}^n \mapsto \{0,1\}$ lies within distance 2^{-n}. Of course, adding more functions to $\mathcal{F}^0_n$ and $\mathcal{F}^1_n$ cannot change this, so let $\mathcal{F}'_n = \mathcal{F}^0_n \cup \mathcal{F}_n$ and $\mathcal{F}''_n = \mathcal{F}^1_n \cup \mathcal{F}_n$ for a family of sets of functions $\{\mathcal{F}_n\}$ as in theorem 4.1. Since the intersection of $\mathcal{F}'_n$ and $\mathcal{F}''_n$ is exactly $\mathcal{F}_n$, the theorem follows. □

Since $\mathcal{F}_n$ as constructed in theorem 4.1 is sparse in $\{\{0,1\}^n \mapsto \{0,1\}\}$, it seems reasonable that doing some kind of trick analoguos to that in theorem 5.2 should yield a similar result for the complement of a trivially testable property. This is indeed the case, although we have to tweak theorem 4.1 a bit.

Theorem 5.3. *There exists a family of sets of functions $\{\mathcal{G}_n\}$ trivially testable for $\epsilon \geq \frac{1}{2^n - 1}$ such that $\{\mathcal{F}_n\}$ defined by $\mathcal{F}_n = \{\{0,1\}^n \mapsto \{0,1\}\} \setminus \mathcal{G}_n$ is not testable in subexponential complexity.*

It is not too hard to guess that what we want to do is to define $\mathcal{G}_n$ as $\{\{0,1\}^n \mapsto \{0,1\}\} \setminus \mathcal{F}_n$ and then once again appeal to theorem 4.1. In order to do this, however, we need to show that for $f \in \mathcal{F}_n$ there is always a $g \notin \mathcal{F}_n$ which is close to f (regardless of the distribution).

Lemma 5.4. *The family of sets of functions $\{\mathcal{F}_n\}$ in theorem 4.1 can be chosen so that for every n and every $f \in \mathcal{F}_n$, there exists at most one function $f' \in \mathcal{F}_n$ such that f and f' differ on exactly one input.*

Proof of lemma 5.4. Let $N = 2^n$. The probability for a fix f that $f \in \mathcal{F}_n$ is $2^{\frac{1}{10}N}/2^N = 2^{-\frac{9}{10}N}$. It follows that the probability that there exist two functions $f_1, f_2 \in \mathcal{F}_n$ such that both f_1 and f_2 differ from this f on exactly one input is bounded by

$$\{\# \text{ such } f_1, f_2\} \cdot \Pr[f_1 \in \mathcal{F}_n \text{ and } f_2 \in \mathcal{F}_n] \lessapprox N^2 \cdot \left(2^{-\frac{9}{10}N}\right)^2 < 2^{-\frac{17}{10}N}$$

and the probability that this event occurs for some $f : \{0,1\}^n \mapsto \{0,1\}$ is at most $2^N \cdot 2^{-\frac{17}{10}N} = 2^{-\frac{7}{10}N}$. Adding this bound to the bound on the probability that $\mathcal{F}_n$ is not hard to test derived in theorem 4.1, we see that there must exist an $\mathcal{F}_n$ which both possesses the property in the statement of the lemma and is hard to test. $\square$

Proof of theorem 5.3. Let $\{\mathcal{F}_n\}$ be as in lemma 5.4 and let $\mathcal{G} = \{\mathcal{G}_n\}$ be defined by $\mathcal{G}_n = \{\{0,1\}^n \mapsto \{0,1\}\} \setminus \mathcal{F}_n$. We are home if we can show that $\mathcal{G}$ is trivially testable.

For each $f \notin \mathcal{G}_n$ (i.e. $f \in \mathcal{F}_n$), there are at least $2^n - 1$ functions in $\mathcal{G}_n$ that differ from f on exactly one input. This implies that there is always a function $g \in \mathcal{G}_n$ which is at distance at most $1/(2^n - 1)$ from f regardless of the distribution $\mathcal{D}_n$ on $\{0,1\}^n$. This is true since the total probability mass of all $x \in \{0,1\}^n$ for which there exists a $g \in \mathcal{G}_n$ such that g and f differ only on x is at most 1, and there are at least $2^n - 1$ such x.

Consequently, as long as $\epsilon \geq \frac{1}{2^n-1}$ the trivial algorithm which accepts all functions is a tester for $\mathcal{G}$. $\square$

6 Some Good News: Bipartiteness

In their article, Goldreich, Goldwasser and Ron present property testers with query complexity $\text{poly}(1/\epsilon)$ for k-colourability, k-clique, k-cut and k-bisection. We present only the special case of 2-colourability, a.k.a. bipartiteness.

6.1 Notation

We consider undirected, simple graphs $G = (V, E)$ (no multiple edges or self-loops), where $V = V(G)$ is the set of vertices of G and $E = E(G)$ is the set of edges. We let $N = |V(G)|$ and write $V(G) = \{1, 2, \ldots, N\}$. In the following, we will use u, v, w to denote arbitrary vertices and S, U, X to denote arbitrary sets of vertices. Unless otherwise stated, V is implicitly assumed to denote $V(G)$.

From the point of view of a property tester, the graph is represented by the indicator function f_G for its edges, i.e.

$$f_G(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

Our testing algorithm A uses queries and works under the uniform distribution. Accordingly, the distance $d(G_1, G_2)$ between two N-vertex graphs G_1 and G_2

is defined as

$$d\big(G_1, G_2\big) = \frac{\big|\{u, v \in V : f_{G_1}(u, v) \neq f_{G_2}(u, v)\}\big|}{N^2},$$

which can be seen to be the fraction (over N^2) of vertex pairs that are adjacent in one graph but not in the other. If $\mathcal{G}$ is a set of graphs, the distance of G from $\mathcal{G}$ is

$$d\big(G, \mathcal{G}\big) = \min_{G' \in \mathcal{G}} \big\{d\big(G, G'\big)\big\}.$$

The set of *neighbours* of v is

$$\Gamma\big(v\big) = \big\{u : (u, v) \in E(G)\big\}$$

and for a set of vertices U we define

$$\Gamma\big(U\big) = \bigcup_{v \in U} \Gamma\big(v\big).$$

The *degree* of v is

$$\deg(v) = \big|\Gamma\big(v\big)\big|.$$

The set of edges between V_1 and V_2 is

$$E\big(V_1, V_2\big) = \big\{(v_1, v_2) \in E(G) : v_1 \in V_1,\ v_2 \in V_2\big\}.$$

A *partition* (U_1, U_2) of U is a pair of sets U_1, U_2 such that $U_1 \cup U_2 = U$ and $U_1 \cap U_2 = \emptyset$ (i.e. $U_1 \mathbin{\dot\cup} U_2 = U$ for $\dot\cup$ the disjoint union).

6.2 The Property Tester for Bipartiteness

An N-vertex graph $G = (V, E)$ is *bipartite* if there is a partition (V_1, V_2) of V such that $E\big(V_1, V_1\big) = E\big(V_2, V_2\big) = \emptyset$.

In view of our distance function, G is ϵ-far from bipartite if and only if for all partitions (V_1, V_2) of V it holds that $\big|E\big(V_1, V_1\big) \cup E\big(V_2, V_2\big)\big| \geq \epsilon N^2$. That is, however we try to partition V into V_1 and V_2, there will always be at least a constant fraction of edges between vertices within V_1 or between vertices within V_2 violating the bipartiteness property. Equivalently, G is ϵ-far from bipartite if we have to remove at least ϵN^2 edges from G to get a bipartite graph.

Note that in particular this means that the only potentially hard graphs for a tester (and thus the only potentially interesting graphs for us) are *dense* graphs with $\Omega\big(N^2\big)$ edges. For a graph with $\mathrm{o}\big(N^2\big)$ edges the tester can always accept, since such a graph is ϵ-close to being bipartite by definition.

The idea behind the property tester is really simple: pick a random subgraph, check if it is bipartite and guess that the whole graph is bipartite if the sample subgraph is.

Definition 6.1 (Bipartiteness property tester). Given $n = N^2 \log N, \epsilon, \delta$ and oracle access to f_G for an unknown N-vertex graph G, do the following.

1. Pick a uniformly random $X \subseteq V(G)$ of size $\big|X\big| = \Theta\big(\log\big(\frac{1}{\epsilon\delta}\big)/\epsilon^2\big)$.

2. For all $u, v \in X$, query if $(u, v) \in E(G)$. Let $G_X = \big(X, E\big(X, X\big)\big)$ be the induced subgraph.

3. Decide whether G_X is bipartite or not by doing a breadth-first search.

4. Accept G if G_X is bipartite and reject otherwise.

Note that the size of the sample X depends on ϵ and δ *but is independent of the graph size* N.

We spend the remaining part of this lecture proving the following result.

Theorem 6.2. *The algorithm in definition 6.1 is a property tester for bipartiteness whose query complexity and running time are* $\mathrm{O}\bigl(\log^2\bigl(\frac{1}{\epsilon\delta}\bigr)/\epsilon^4\bigr)$. *A bipartite graph* G *is accepted with probability* 1*, and if* G *is* ϵ*-far from being bipartite it is rejected with probability at least* $1-\delta$ *(over the coin tosses of the algorithm).*

Before diving into the proof, let us make a short detour to argue why this result is quite remarkable.

As was noted in the introduction, the notion of testing a graph property P is a relaxation of the notion of deciding (recognizing) the graph property P. In the classical decision problem, there are no margins of error, and one is required to accept all graphs having property P and reject all graphs which lack it. It has been shown that any deterministic procedure for deciding any nontrivial monotone N-vertex graph property must examine $\Omega\bigl(N^2\bigr)$ entries in the adjacency matrix representing the graph. For randomized decision procedures, an $\Omega\bigl(N^{4/3}\bigr)$ lower bound for the query complexity has been shown (see the article by Goldreich, Goldwasser and Ron for references).

This stands in striking contrast to the results for bipartiteness, where we can test for the property by examining a *constant number of entries* in the adjacency matrix!

On the other hand, one could argue that this striking result to some extent is a consequence of the fact that the distribution and the associated distance function are chosen the way they are. As was noted above, for graphs with $\mathrm{o}\bigl(N^2\bigr)$ edges it is always in order to accept, and for graphs with $\Omega\bigl(N^2\bigr)$ edges one could expect that it should be easy to find a short cycle of odd length showing that the graph is not bipartite.

6.3 Analysis of the Tester: Proof Idea

It is clear that the tester has query and time complexity as stated in the theorem. It is also clear that a bipartite graph is always accepted by the tester. Assume therefore that the graph G is ϵ-far from being bipartite. We need to show that G is accepted with probability at most δ.

The idea of the proof is as follows.

1. For purposes of analysis only, let us assume that our sample of vertices is in fact chosen as $X = U \,\dot\cup\, S$. We want to use U to induce candidate partitions (V_1, V_2) of V, and S to evaluate whether these candidate partitions are good.

2. A partition (U_1, U_2) of U can be seen as the embryo of a partition of the whole graph G in the natural way. Very roughly, the vertices in $\Gamma\bigl(U_1\bigr)$ and $\Gamma\bigl(U_2\bigr)$ should be on opposite sides, and any vertices in $V \setminus \Gamma\bigl(U\bigr)$ may be placed on either side. We consider the partitions induced on G in (almost) this way for all partitions (U_1, U_2) of U, and hope that the set $V \setminus \Gamma\bigl(U\bigr)$ is

small so that we can get a lot of information about (V_1, V_2) just by looking at (U_1, U_2).

3. Now if the graph is ϵ-far from bipartite, for every partition (V_1, V_2) there will be many edges violating the bipartiteness property (at least ϵN^2 of them). Since these edges are all over the place, if we picked our "evaluation set" S of the right size, chances are that some of these violating edges will go between vertices in S.

4. But if this holds for all partitions (U_1, U_2) of U, there can be no bipartite partition $(U_1 \cup S_1, U_2 \cup S_2)$ of X. That is, G_X is not bipartite, and the tester will discover this and reject.

6.4 The Proof

We write our uniformly random sample $X \subseteq V(G)$ as $X = U \mathbin{\dot\cup} S$ for

$$|U| = t = \frac{3}{\epsilon} \log\left(\frac{6}{\epsilon\delta}\right) \tag{12}$$

and

$$|S| = m = \frac{6t}{\epsilon} + 2t. \tag{13}$$

Then X has the right size $|X| = \Theta\big(\log(\frac{1}{\epsilon\delta})/\epsilon^2\big)$.

Remark 6.3. It follows from the above that U can be seen as selected uniformly at random from V and S as selected uniformly at random from $V \setminus U$. However, in the proof of lemma 6.9 below it is assumed that the vertices in S are selected independently one by one *with replacement* (or in fact two by two with replacement) in order to make the calculations simpler.

The authors of the paper mention this simplification explicitly and encourage "the formally-inclined reader" to work out the necessary details to fix this. Again, it seems this can be done by proving that if U and S are chosen as multisets, almost surely they will be disjoint and of basically the right sizes, and again we leave the details to the reader.

Definition 6.4 (Violating edges and good partitions). An edge (u, v) is a *violating edge* with respect to the partition (V_1, V_2) of V if $u, v \in V_1$ or $u, v \in V_2$.

If a partition (V_1, V_2) of V has at most ϵN^2 violating edges it is *ϵ-good*, otherwise it is *ϵ-bad*. A partition without violating edges is *perfect*.

Note that if there is a perfect partition (V_1, V_2) of V, the graph G is bipartite. If G is ϵ-far from bipartite every partition is ϵ-bad.

Definition 6.5 (Influential vertex). A vertex $v \in V(G)$ is *influential* if $\deg(v) \geq \frac{\epsilon}{3}N$.

We will focus most of our attention on the influential vertices, since they contribute most of the violating edges. There are at most $N \cdot \frac{\epsilon}{3}N$ edges in total incident to non-influential vertices, so by necessity at least $\frac{2\epsilon}{3}N^2$ violating edges in an ϵ-bad partition emanate from influential vertices.

We consider U a good sample if most of the influential vertices are neighbours of vertices in U.

Definition 6.6 (Well-covering set). The certex set U *covers* the vertex v if $v \in \Gamma(U)$. U is a *well-covering set* if at most $\frac{\epsilon}{3}N$ of the influential vertices are *not* covered by U.

We now have all but one of the definitions that we will need. Before giving the actual proof, we relate the proof idea sketched above to our new definitions. We have two goals.

1. We want to show that a randomly chosen U is very likely to be well-covering. This means that a partition (U_1, U_2) of U tells us a lot about the influential vertices in G, which as noted above are the ones responsible for a majority of the violating edges.
2. Given that U is well-covering and that G is ϵ-far from bipartite, we want to show that however U was partitioned into (U_1, U_2), when we choose S it is very unlikely that $U \dot\cup S$ can be partitioned as $(U_1 \dot\cup S_1, U_2 \dot\cup S_2$ in any consistent way. In other words, G_X will very likely not be bipartite, and the tester will reject G.

The partition (U_1, U_2) of U does not help us to analyze vertices that are not covered by U. Thus we start by ensuring that most vertices are covered by U, at least most influential ones.

Lemma 6.7. *With probability at least $1 - \delta/2$, a uniformly chosen U of size $t = \frac{3}{\epsilon}\log\left(\frac{6}{\epsilon\delta}\right)$ is a well-covering set for $V(G)$.*

Proof. Suppose that v is influential. Then

$$
\begin{aligned}
\Pr_U\left[v \notin \Gamma(U)\right] &\le \frac{\binom{N-\frac{\epsilon}{3}N}{t}}{\binom{N}{t}} \\
&\le \left(\frac{N\left(1-\frac{\epsilon}{3}\right)}{N}\right)^t && \text{[by fact 3.5]} \\
&\le \exp\left(-\frac{\epsilon}{3}t\right) && \text{[use } 1+x \le e^x\text{]} \\
&\le \frac{\epsilon\delta}{6} && (14)
\end{aligned}
$$

and it follows that the expected number of influential vertices without neighbours in U can be at most $\frac{\epsilon\delta}{6}N$. By Markov's inequality, the probability that the expected value is exceeded by a factor of $2/\delta$ is at most $\delta/2$. $\square$

If (U_1, U_2) is a partition of a well-covering set U, we can concentrate on analyzing violating edges in $E\left(\Gamma(U_i), \Gamma(U_i)\right)$ for $i = 1, 2$. This is so since the total number of edges incident to vertices *not* in U is small. We say that a partition (U_1, U_2) of U is ϵ-*useful* if there are few such violating edges. Here "useful" is to be understood in the sense of "potentially useful for constructing a good partition for $V(G)$". In lemma 6.11 we will show that usefulness in fact implies that $V(G)$ has a good partition.

Definition 6.8. A partition (U_1, U_2) of U is ϵ-*useful* if

$$\left|\left\{(v_1, v_2) \in E(G) : v_1, v_2 \in \Gamma(U_1) \text{ or } v_1, v_2 \in \Gamma(U_1)\right\}\right| < \frac{\epsilon}{3}N^2$$

and ϵ-*unusable* otherwise.

If (U_1, U_2) is ϵ-unusable, then with high probability this will show up in the sample S in the form of an edge $(v, v') \in S \times S$ for $v, v' \in \Gamma(U_i)$, say for $i = 1$. Let u, u' be the neighbours in U_1 of v and v', respectively. If $u = u'$, there is a triangle in G_X (which means that G fails the test), and otherwise the path $u \to v \to v' \to u'$ shows that u and u' must belong to different sides of the partition (and so (U_1, U_2) cannot be used in the partition of X). We formalize this.

Lemma 6.9. *If* (U_1, U_2) *is* ϵ*-unusable, then*

$$\Pr_S \left[\forall\, v, v' \in S \left(\bigwedge_{i=1,2} (v, v') \notin E\big(\Gamma(U_i), \Gamma(U_i)\big) \right) \right] < \delta \cdot 2^{-(t+1)}.$$

Proof. If (U_1, U_2) is ϵ-unusable, the inequality in definition 6.8 tells us that

$$\Pr_{v,v'} \left[\bigvee_{i=1,2} (v_1, v_2) \in E\big(\Gamma(U_i), \Gamma(U_i)\big) \right] \geq \frac{\epsilon}{3}.$$

If we construct S by drawing $m/2$ random pairs of distinct vertices (see remark 6.3), we have

$$\begin{aligned}
&\Pr_S \left[\forall\, v, v' \in S \left(\bigwedge_{i=1,2} (v, v') \notin E\big(\Gamma(U_i), \Gamma(U_i)\big) \right) \right] \\
&\leq \left(1 - \frac{\epsilon}{3}\right)^{m/2} \\
&\leq \exp\left(-\frac{\epsilon}{6} m\right) \\
&= \exp\left(-\left(t + \frac{\epsilon}{3} t\right)\right) \\
&= \frac{\epsilon\delta}{6} \cdot e^{-t} \\
&\leq \delta \cdot 2^{-(t+1)}
\end{aligned} \tag{15}$$

and the lemma follows. ☐

This lemma immediately yields a highly usable corollary.

Corollary 6.10. *If every partition* (U_1, U_2) *of* U *is* ϵ*-unusable, then for a uniformly chosen* S *it holds with probability at least* $1 - \delta/2$ *that there is no perfect partition of* $X = U \,\dot{\cup}\, S$.

Proof. By the union bound, the probability that there is a perfect partition of X although all 2^t partitions of U are ϵ-unusable is at most

$$\begin{aligned}
&\sum_{(U_1,U_2)} \Pr_S \left[\forall\, v, v' \in S \left(\bigwedge_{i=1,2} (v, v') \notin E\big(\Gamma(U_i), \Gamma(U_i)\big) \right) \right] \\
&\leq\; 2^t \cdot \delta \cdot 2^{-(t+1)} \;=\; \delta/2.
\end{aligned}$$

☐

This means that G_X is found to be bipartite, and the tester rejects G.

Now we are almost finished. The final piece needed to put the proof together is the next lemma.

Lemma 6.11. *If there is a well-covering set U of size t for the graph G which has an ϵ-useful partition (U_1, U_2), then G is ϵ-close to bipartite.*

Intuitively, the reason for this is that there simply are not enough edges left outside $E(\Gamma(U), \Gamma(U))$ to violate in order to make G ϵ-far from bipartite. Before proving this lemma, however, let us see why theorem 6.2 now follows.

Proof of theorem 6.2. As noted above, the only thing in the theorem that is not immediately clear is that a graph G which is ϵ-far from bipartite is accepted with probability at most δ. Let us assume that G is ϵ-far and estimate the probability of error. We have

$$\begin{aligned}\Pr_X[\text{accept}] = \; & \Pr_S[\text{accept} \mid U \text{ well-covering}] \cdot \Pr_U[U \text{ well-covering}] \\ & + \Pr_S[\text{accept} \mid U \text{ not well-covering}] \cdot \Pr_U[U \text{ not well-covering}].\end{aligned}$$

By lemma 6.7, $\Pr_U[U \text{ not well-covering}] \leq \delta/2$. Lemma 6.11 informs us that all partitions of a well-covering set U are ϵ-unusable, and combining this with corollary 6.10 we have $\Pr_S[\text{accept} \mid U \text{ well-covering}] \leq \delta/2$. Thus

$$\Pr_X[\text{accept}] \;\leq\; \delta/2 \cdot (1 - \delta/2) + 1 \cdot \delta/2 \;\leq\; \delta$$

and theorem 6.2 follows. □

It remains to prove the final lemma.

Proof of lemma 6.11. We choose the partition (V_1, V_2) of V as $V_1 = \Gamma(U_2)$, $V_2 = V \setminus V_1$ and count the number of violating edges with respect to (V_1, V_2).

1. Edges incident to non-influential vertices: There are at most N such vertices and at most $\frac{\epsilon}{3}N$ edges from each such vertex, for a total of at most $\frac{\epsilon}{3}N^2$ edges.

2. Edges incident to influential vertices not covered by U: There are at most $\frac{\epsilon}{3}N$ such vertices and at most N edges from each such vertex, again at most $\frac{\epsilon}{3}N^2$ edges.

3. Violating edges incident to neighbours of U: These are edges (v, v') with $v, v' \in \Gamma(U)$. We have two cases.

   (a) Edges $(v, v') \in E(V_1, V_1)$: By construction $v, v' \in \Gamma(U_2)$.

   (b) Edges $(v, v') \in E(V_2, V_2)$: Again by construction we know that $v, v' \notin \Gamma(U_2)$ (otherwise they would be members of V_1), but $v, v' \in \Gamma(U)$ so it must be the case that $v, v' \in \Gamma(U_1)$.

   Since (U_1, U_2) is ϵ-useful, the total number of such edges is at most $\frac{\epsilon}{3}N^2$.

We see that the total number of violated edges is at most $3 \cdot \frac{\epsilon}{3}N^2 = \epsilon N^2$, i.e. the partition (V_1, V_2) of V is ϵ-good, and so G is ϵ-close to bipartite. □

6.5 Some Final Remarks

We have three final remarks.

1. If G is a bipartite graph, the partition of $X = U \mathbin{\dot\cup} S$ can be shown to induce an ϵ-good partition (V_1, V_2) of V with probability at least $1 - \delta$ (over the choice of X). So not only can we test for bipartiteness, we can also efficiently construct an approximate partition of the vertices.

2. One can save a factor $1/\epsilon$ in the query complexity and running time of the bipartiteness property tester to get $\mathrm{O}\big(\log^2\big(\frac{1}{\epsilon\delta}\big)/\epsilon^3\big)$. This is done by observing that the pairwise chosen edges $(v_1, v_1'), (v_2, v_2'), \ldots$ in the proof of lemma 6.9 are the only edges in $S \times S$ that we actually look at. We do not take into consideration in our analysis whether, say, the edge (v_1, v_2') is present or not. So we can pick U and S as above and then query all pairs of vertices in U but only the $m/2$ vertex pairs for S in the proof of lemma 6.9.

3. A natural question is whether we need to be able to query f_G to get a good bipartiteness tester or if it is possible to use samples. We conclude these notes by stating a theorem saying (basically) that efficient testing based on samples only is not possible.

Theorem 6.12. *Any property testing algorithm for bipartiteness that observes only a random labelled sample must have sample complexity* $\Omega\big(\sqrt{N}\big)$.

Proof. Looks like this is going to be part of the homework... □