Randomness and Computation 2018/19
Week 7 tutorial sheet (12-1pm, Tues 5th, Wed 6th March)

1. Vertex cover question.

   (a) The simplest example of this poor result is the scenario of a star graph G where we have one central vertex s and $n-1$ other vertices, and $E = \{(s, v); v \in V \setminus \{s\}\}$.
   Then when the algorithm chooses an "uncovered edge" $e = (s, v)$, imagine the algorithm *always* takes the non-centre vertex v as the arbitrary endpoint. Then we only have "covered" one extra edge with this addition. With this strategy, the algorithm will eventually uncover all $n-1$ edges with a set C of size $n-1$. However, the size of the optimal cover is 1. Hence we build a cover $\Omega(n)$ times the size of the optimal one.

   (b) We have now modified the algorithm so that when it considers an (arbitrary) uncovered edge $e = (u, v)$, it tosses a fair coin to add u or v with probability 1/2 each. To see that this ensures a good expected size for the eventual cover, we will need some definitions.
   Suppose that the minimum cover size is k for our given graph G. Let X_k be the random variable for the number of steps(/vertex additions) of the algorithm needed to cover the whole graph. We want to bound E[X_k]. It makes sense to generalise this variable to represent "mid-algorithm" scenarios such as when we have some covered edges, and know that some pool of vertices of size ℓ suffices to cover all of the *remaining uncovered edges*. Let X_ℓ denote the (random) number of vertices that will be added to complete the cover, in this case.
   We will examine the effect on cover size of a single step of the algorithm for a graph which needs $\ell, \ell \geq 1$ nodes to finish covering its remaining edges. In step 2., we take an arbitrary uncovered edge (u, v). Note that in the underlying best cover for the remaining edges, *at least one* of u, v belongs to that optimum cover. Assume wlog that is u.
   Whichever of u, v are chosen in step 2, we will have a reduced number of uncovered edges (*at least* one extra edge has been covered by the added vertex, possibly more). If u was added (prob. 1/2), there is a set of $\ell - 1$ vertices that can be used to finish covering the "still uncovered" edges of the graph. If v was instead added (prob. 1/2), then there is certainly still a set of ℓ vertices that would cover the remaining uncovered edges (we certainly cannot have made the situation worse by adding a vertex). Hence,

   $$X_\ell \;=\; 1 + \frac{1}{2}(X'_\ell + X'_{\ell-1}),$$

   where X'_ℓ and $X'_{\ell-1}$ are used to denote the fact that the graph (or at least the collection of uncovered edges) have changed a bit. Hence

   $$X_\ell \;-\frac{1}{2}X'_\ell = \;1 + \frac{X'_{\ell-1}}{2},$$

   and also we know that $X'_\ell \leq X_\ell$ as any sequence of choices that might cover the uncovered edges before step 2, *definitely* also covers the uncovered edges after. Hence we know

   $$\frac{X_\ell}{2} \leq 1 + \frac{X'_{\ell-1}}{2}, \text{ that is, } X_\ell \leq 1 + X'_{\ell-1}.$$

Now the $'$ on the $X'_{\ell-1}$ is irrelevant, as we didn't assume anything about the structure of the graph/uncovered edges in our argument, only that there was a completion-cover of size ℓ. Hence we know that $E[X_\ell] \leq 2 + E[X_{\ell-1}]$. Applying this iteratively for $\ell = k, \ell = k-1, \ell = k-2, \ldots, \ell = 1$, we find that $E[X_k] \leq 2 \cdot k$, which is just twice the size of the optimal cover, as required.

(c) We first show that $\Pr[X_k \leq 2k] \geq \frac{1}{n}$ for a single run of our algorithm, assuming $n \geq 3$ (a reasonable assumption). We will show this (in some ways similar to the proof of Markov's Inequality) by proving by contradiction. It is possible to show other lower bounds on $\Pr[X_k \leq 2k]$ such as $\Pr[X_k \leq 2k] \geq \frac{1}{2k}$ (using similar arguments), or even $\frac{1}{n-1-k}$ but it's harder to make use of those when we won't *know* the value of k in advance so I am not bothering to write those details out.

Suppose instead that it was the case that $\Pr[X_k \leq 2k] < \frac{1}{n}$. Consider how the value of $E[X_k]$ depends on the probabilities $\Pr[|X_k| \leq 2k]$ and $\Pr[|X_k| > 2k]$ and notice that certainly

$$E[|X_k|] \ \geq k \cdot \Pr[|X_k| \leq 2k] + (2k+1) \cdot \Pr[|X_k| > 2k],$$

the factor k appearing because this is the least possible size of a vertex cover in the range $1, \ldots, 2k$ and the $(2k+1)$ being the least possible value in the range $2k+1, \ldots, \frac{n}{2}$. So if it was the case that $\Pr[X_k \leq 2k]$ was less than n^{-1}, then we would have

$$\begin{aligned} E[|X_k|] &> k \cdot \frac{1}{n} + (2k+1) \cdot (1 - \frac{1}{n}) \\ &= \frac{k}{n} + (2k+1) - \frac{2k}{n} - \frac{1}{n} \\ &= 2k + 1 - \frac{k}{n} - \frac{1}{n} \\ &= 2k + 1 - \frac{1}{2} - \frac{1}{n} \\ &> 2k, \end{aligned}$$

contradicting our assumption of $E[|X_k|] \leq 2k$. Note we assumed that $k < \frac{n}{2}$ in the second-last step, as otherwise $2k \geq n$ in which case *all* vertex covers satisfy the $\leq 2k$ condition. In the final step we are using the $n \geq 3$ assumption.

Therefore we have shown that $\Pr[|X_k| \leq 2k] \geq \frac{1}{n}$ when we run the algorithm once and take that vertex cover.

Now imagine we run the algorithm ℓ times, generating ℓ different vertex cover sets and taking the best (smallest) such set as our overall result. We now evaluate our probability of returning a cover of size $\leq 2k$.

On a single run of the randomised-endpoint algorithm of (b), the probability that we fail to return a cover of size $\leq 2k$ is $\leq (1 - \frac{1}{n})$. Therefore, over ℓ repeated experiments, the probability that we fail to get *any* cover of size $\leq 2k$ is at most $(1-\frac{1}{n})^\ell$. Now $(1-\frac{1}{r})^r < e^{-1}$ for all $r \in \mathbb{N}, r \geq 2$ (working from those standard inequalities $(1 + \frac{1}{r})^r < e < (1 + \frac{1}{r})^{r+1}$) hence if we set ℓ to be some factor of n we will have a failure probability below e^{-1}.

We have a failure probability of at most $(1-\frac{1}{n})^\ell < e^{-\ell/n}$, and we need to set ℓ high enough to ensure that $e^{-\ell/n} \le \delta$. This is equivalent to asking

$$\begin{array}{rcll} \delta^{-1} & \le & e^{\ell/n} & \Leftrightarrow \\ \ln(\delta^{-1}) & \le & \ell/n, & \end{array}$$

or in other words, that $\ell \ge n\ln(\delta^{-1})$.

note: In my original setting of this question, I used the symbol k in the statement of (c) to mean "number of runs of the algorithm". I should never have done that, as it was already used to denote the size of the best vertex cover. The tutorial sheet is changed to now show ℓ instead.

(d) For this question, consider the same example as in (a), but assign the weight Mn to the central vertex, for some very large value of M. Let all the vertices $v \in V \setminus \{s\}$ have weight 1. In this case the optimal weighted vertex cover is now the set $V \setminus \{s\}$. However, with probability $1 - \frac{1}{2^{n-1}}$ the vertex cover computed by our modified algorithm (of (b)) will contain vertex s, and hence have weight $\ge M \cdot n$. Note that we can make this approximation as bad as we want by increasing M.

(e) We have altered the (randomised) algorithm of part (b), and while we still choose an edge arbitrarily, once we have this edge $e = (u, v)$, we then add u with probability $\frac{w(v)}{w(u)+w(v)}$ and vice versa for v. Optimum vertex cover has weight W.

We generalise our (b) definitions to let X_W be the random variable for the total weight of the vertices added by the Algorithm to cover the whole graph, assuming the "best" cover has weight W. For any value $U \le W$, we will use the random variable X_U for "mid-algorithm" scenarios when we have some covered edges, and know that some pool of vertices of total weight U suffices to cover all of the *remaining uncovered edges*. X_U will be the weight of all vertices added by the algorithm in completing the cover.

We will examine the effect on cover weight of a single step of the algorithm in the situation where the best set of vertices to finish covering the graph has weight U. In step 2., we take an arbitrary uncovered edge (u, v). Note that in the underlying best cover for the remaining edges, *at least one* of u, v belongs to that optimum cover. Assume wlog that is u.

Whichever of u, v are chosen in step 2, we will have a reduced number of uncovered edges (*at least* one extra edge has been covered by the added vertex, possibly more). If u was added (prob. $\frac{w(v)}{w(u)+w(v)}$), there is a set of vertices of weight $U - w(u)$ that can be used to finish covering the "still uncovered" edges of the graph. If v was instead added (prob. $\frac{w(u)}{w(u)+w(v)}$), then the original set of vertices of total weight U will certainly cover the remaining uncovered edges (we cannot have made the situation worse by adding a vertex). Hence,

$$X_U = \frac{w(v)}{w(u)+w(v)} \cdot (w(u) + X'_{U-w(u)}) + \frac{w(u)}{w(u)+w(v)} \cdot (w(v) + X'_U),$$

where X'_U and $X'_{U-w(u)}$ are used to denote the fact that the graph (or at least the collection of uncovered edges) has changed a bit. Hence

$$X_U = \frac{2w(u)w(v)}{w(u)+w(v)} + \frac{1}{w(u)+w(v)} \cdot (w(u)X'_U + w(v)X'_{U-w(u)}), \text{ ie}$$

$$X_U - \frac{w(u)}{w(u)+w(v)}X'_U = \frac{2w(u)w(v)}{w(u)+w(v)} + \frac{w(v)}{w(u)+w(v)} \cdot X'_{U-w(u)}.$$

We know that $X'_U \leq X_U$ as any sequence of choices that might cover the uncovered edges before step 2, *definitely* also covers the uncovered edges after. Hence it is certainly the case that

$$\frac{w(v)}{w(u)+w(v)} \cdot X_U \leq \frac{2w(u)w(v)}{w(u)+w(v)} + \frac{w(v)}{w(u)+w(v)}X'_{U-w(u)}, \text{ ie,}$$

$$X_U \leq 2w(u) + X'_{U-w(u)}.$$

Now the $'$ on the $X'_{U-w(u)}$ is irrelevant, as we didn't assume anything about the structure of the graph/uncovered edges in our argument, only that there was a completion-cover of size ℓ. Hence we know that $\mathrm{E}[X_U] \leq 2w(u) + \mathrm{E}[X_{U-w(u)}]$. Applying this iteratively for the various "cover vertices" (each uncovered edge choice always has one endpoint which is a "cover vertex" for the solution of optimal value) we find that $\mathrm{E}[X_W] \leq W$, just twice the size of the optimal cover, as required.

2. We consider various structures in the $G_{n,p}$ model and ask how $p = p(n)$ should be set if we want to ensure that the *expected number* of the particular structure will be exactly 1. Exercise 5.16 of [MU].

   Note that all solutions can make use of the *linearity of expectation*, by considering the probability that an fixed group of vertices (or maybe edges) has this structure, and then multiplying by the number of different ways to choose the group of vertices (or edges, in the case of (c)).

   (a) For any specific 5 vertices in a graph, we will only have a a clique if all of the potential edges between these vertices belong to the graph. There are $\binom{5}{2} = 10$ possible edges between those 5 vertices, so the probability that these are added by the $G_{n,p}$ process is exactly p^{10} (we don't care what happens with any potential edges outside this group).

   There are $\binom{n}{5}$ ways to choose a group of 5 vertices from the n available, hence the *expected number of* 5*-cliques* for $G \in G_{n,p}$ is

   $$p^{10}\frac{n(n-1)(n-2)(n-3)(n-4)}{5!}.$$

   This expectation will be exactly 1 if we have $p(n) =_{\mathrm{def}} (\frac{120}{n(n-1)(n-2)(n-3)(n-4)})^{0.1}$.

(b) For any specific 6 vertices of a graph, with 3 designated "right" and another three designated "left", they will only form a $K_{3,3}$ if they contain the 9 defining edges across the bipartition, and *fail to contain* the 3 edges on the left-hand side, and *fail to contain* the 3 edges on the right-hand side.

The probability of this edge arrangement in $G_{n,p}$ for such a group of 6 (already bipartitioned into left and right) is $p^9(1-p)^6$.

Now consider how many ways we can construct a pair of these "groups of 3". We can choose the left-3 in $\binom{n}{3}$ ways, the right-3 in $\binom{n-3}{3}$ ways. This is $\frac{n(n-1)(n-2)(n-3)(n-4)(n-5)}{6^2}$. Also note that the definition of $K_{3,3}$ doesn't designate vertices as being "left" or "right", so we must divide by 2 to get the following number of *potential* copies of $K_{3,3}$ in the graph. Hence, taking the probability for a particular pair of "groups of 3", by the number of possible vertex collections, the *expected number of* $K_{3,3}$ *copies is*

$$p^9(1-p)^6 \frac{n(n-1)(n-2)(n-3)(n-4)(n-5)}{72}.$$

To get this exactly equal to 1, we would require p to satisfy

$$p^3(1-p)^2 = \left(\frac{72}{n(n-1)(n-2)(n-3)(n-4)(n-5)}\right)^{1/3}.$$

Roughly (and yes it's not exact), we require *either*

$$p \sim \left(\frac{72}{n(n-1)(n-2)(n-3)(n-4)(n-5)}\right)^{1/9}$$

or

$$p \sim 1 - \left(\frac{72}{n(n-1)(n-2)(n-3)(n-4)(n-5)}\right)^{1/6}.$$

We can separate these because the target on the rhs is small (for reasonably big n), so if p is close to this, we know $(1-p)$ is almost 1, and vice versa. However, we do have these two cases for the range of p.

(c) Finally, for Hamiltonian cycles in G, we are interested in finding a cycle of length n, with edges between each adjacent pair. For a particular ordering of the n vertices, the HC edges will be present with probability p^n (which may be very small).

However, there are many possible orderings of the vertices; we have $n!$ permutations of the vertices in total, and dividing by $2n$ (since rotations or reversals don't change the HC), this is $\frac{(n-1)!}{2}$. So our question is how to set p to achieve $p^n \cdot \frac{(n-1)!}{2}$ to be 1. Using Stirling's formula, we know $(n-1)! \sim \sqrt{2\pi(n-1)}(\frac{n-1}{e})^{n-1}$, hence we need $p \sim \frac{e}{n-1}$.

Note I am using the knowledge that $\sqrt{2\pi(n-1)}^{\frac{1}{n}}$ is close to 1 for reasonably large n here. Not true if n is small.

3. This was question 6.9 of [MU]. We are interested in the question, for a given tournament graph, or whether there exist rankings which are close (in both direction) to matching 50% of the tournament directions.

   We make one observation to start with, which is that for a *uniform random ranking* of the vertices V of the graph $G = (V, \vec{E})$, that the *expected number of arcs* of $\vec{E}$ that agree with the ranking is exactly $\frac{|\vec{E}|}{2}$. This is due to linearity of expectation, and the fact that for any individual arc, the probability that a random ranking has its vertices in the right order is exactly $\frac{1}{2}$.

   (a) As shown already, for any tournament $G = (V, \vec{E})$, $\mathrm{E}_{\pi \in S_n}[|\{(u \to v) \in \vec{E} : \pi(u) < \pi(v)\}|]$ is exactly $\frac{|\vec{E}|}{2}$ (which is exactly $\frac{n(n-1)}{4}$ as it happens, but not important). However, the expectation was calculated over *rankings of the vertices* (ie, permutations of V). If the expected value is exactly $\frac{|\vec{E}|}{2}$ (over all the possible rankings), then there is certainly at least one ranking whose consistency with $\vec{E}$ is *at least as* good as that. Hence (this being a simple example of the probabilistic method) we know at least one ranking consistent with 50% or greater of the arcs.

   (b) We are now asked to show existence of a tournament for which *all possible rankings* disagree with 49% of the arcs of the graph.

   First recall that we have shown for any specific tournament $\vec{E}$ that the expected number of wrongly ordered pairs is exactly $\frac{|\vec{E}|}{2}$ (which is exactly $\frac{n(n-1)}{4}$, given that tournaments have a directed arc for each vertex pair). We can also apply this in the opposite direction, starting from a specific ranking $\pi \in S_n$: if we generate a random tournament $\vec{E}$ from the $2^{\frac{n(n-1)}{2}}$ possible tournaments, then the expected number of disagreements between π and $\vec{E}$ is also $\frac{n(n-1)}{4}$.

   Let us write, for the fixed π as the sum $X = \sum_{k=1}^{n(n-1)/2} X_k$ of $\frac{n(n-1)}{2}$ random variables with parameter $\frac{1}{2}$ (one for each edge of the complete graph). Note $\mu = \mathrm{E}[X]$ is exactly $\frac{n(n-1)}{4}$. We will now apply Chernoff Bounds to X to bound the probability of being a factor of more than $1 \pm \frac{1}{50}$ from μ, using $\delta = \frac{1}{50}$ in the one-sided bound (4.11) which states

$$\Pr[X - \mu \geq \delta\mu] \;\leq\; e^{-\mu\delta^2/3},$$

   and for us, with $\delta =_{\mathsf{def}} \frac{1}{50}$ and $\mu = \frac{n(n-1)}{4}$, this probability on the right is $e^{-\frac{n(n-1)}{30000}}$. Observe that if we take $n \geq (30001)^2$, say, that the exponent of the e will be less than $-n^{3/2}$; hence a very small probability.

   We started with an arbitrary ranking, and there are $n!$ rankings in total. If we want to upper bound the probability that a random tournament disagrees with at least 49% of edges of *some ranking*, we can use the Union Bound (on rankings), and multiply our probability bound by $n!$, to get the overall failure probability bounded by

$$n! \cdot e^{-\frac{n(n-1)}{30000}},$$

which is less than $n!e^{-n^{3/2}}$ for $n \geq (30001)^2$. Then noting that $n!$ is roughly $(\frac{n}{e})^n\sqrt{2\pi n}$ for large n, we can see that for $n \geq 30001$,

$$\ln\left(n!e^{-n^{3/2}}\right) \sim n\ln(n) - n + \tfrac{1}{2}\ln(2\pi n) - n^{3/2},$$

which is very definitely negative. Hence the quantity $n!e^{-n^{3/2}}$ is very definitely less than 1, and hence the probability that a random tournament disagrees by $\leq 49\%$ of edges of *some ranking in* S_n is strictly less than 1.

Hence by the probabilistic method, there is certainly one tournament that will disagree by $\geq 49\%$ of edges with all rankings in S_n.

Mary Cryan, (updated) 17th April