Tournaments

Drew Armstrong
armstron@math.umn.edu

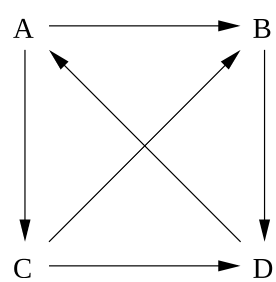
September 29, 2006

Suppose we have a collection of n commercial products or n sports teams and we want to compare them to determine "which is the strongest?" or "which is the weakest?". Ideally, we would like to assign each object a "strength value" that expresses its strength relative to the other objects.

In the "real" world, how would we go about doing this? Perhaps it is very difficult to compare the objects in bunches, but if we compare the objects just two at a time, it is always possible to choose one over the other. Then we would like to compare each pair of objects, choose a winner, and then analyze the results.

Definition 1. A (round-robin) tournament is an orientation of the complete graph K_n.

For example, the following tournament is an orientation of K_4:



We have labelled the vertices by $\{A, B, C, D\}$, and we might interpret these as "brands of toothpaste". The arrow $A \to B$ means that "brand A is judged to be a better toothpaste than brand B". In general,

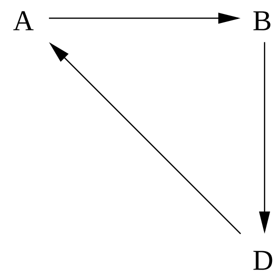
Definition 2. For each directed edge $u \to v$ in a tournament T, we say that u dominates v.

So which toothpaste is best? It is not immediately clear, since there is no toothpaste that dominates every other kind of toothpaste. Our goal is to find a ranking of the vertices from best to worst (say (A, B, C, D)), with the property that

- If $u \to v$, then vertex u should rank higher than vertex v. (It's only fair!)

Definition 3. Suppose we have a ranking of the vertices in a tournament T with v ranked above u, but $u \to v$. This $u \to v$ is called an upset in the ranking. Any ranking without upsets is called a perfect ranking of T.

Thus, our goal should be to find a perfect ranking. But observe that our tournament contains an "oriented triangle":



If we could find a perfect ranking of the tournament, then we would have A ranked higher than B, B ranked higher than D, and D ranked higher than A. Clearly this is impossible. In fact, any oriented cycle in a tournament will cause an upset.

Proposition 4. *If a tournament T contains an oriented cycle, then it has no perfect ranking.*

Proof. Consider an oriented cycle $v_1 \to v_2 \to \cdots \to v_k \to v_1$ in the tournament. Thus in any perfect ranking of T, v_1 must come before v_2, v_2 must come before v_3, and so on. Continuing, we find that v_1 must come before v_k. But since $v_k \to v_1$, v_k must come before v_1. This is a contradiction. Hence T contains no perfect ranking. □

This is quite discouraging. Maybe our goal was too optimistic? It remains to be seen whether directed cycles are very common, or if they are rare. If it turns out that directed cycles are common, we will need to find a new, less optimistic, goal.

Let us examine this issue a bit further.

Definition 5. A tournament T is called transitive if, whenver $u \to v$ and $v \to w$, it follows that $u \to w$. We might also call such a tournament "consistent" since everyone is behaving in a consistent way, without wacky upsets.

Theorem 6. *Given a tournament T, the following statements are equivalent.*

1. *T has a perfect ranking.*

2. *T is transitive.*

3. *T contains no oriented cycle.*

Before we prove this, we need to establish an important technical property of tournaments. We say that a path in a graph G is a spanning path if it covers every vertex in the graph (however, it may omit some edges).

Lemma 7. *Every tournament T possesses a directed spanning path.*

Proof. We prove this by induction on the number of vertices n. If $n = 2$, then certainly our tournament has a spanning path. Now suppose that every tournament on $n-1$ vertices has a spanning path, and consider an arbitrary tournament T on n vertices. If we delete some arbitrary vertex v, then we are left with a tournament T' on $n-1$ vertices, and by induction, T' contains a spanning path $v_1 \to v_2 \to \cdots \to v_{n-1}$.

Now consider the tournament T again. If $v \to v_1$ or $v_{n-1} \to v$, then we are done since $vv_1v_2\cdots v_{n-1}$ or $v_1v_2\cdots v_{n-1}v$ is a spanning path, respectively. So suppose that $v_1 \to v$ and $v \to v_{n-1}$, and consider the smallest i such that $v \to v_i$. (Note that $1 < i \leq n-1$.) Since i is the smallest, we must have $v_{i-1} \to v$, and then

$$v_1v_2\cdots v_{i-1}vv_i\cdots v_{n-1}$$

is a spanning path for T. □

Now we can prove Theorem 6.

Proof. We will show that (1) ⇒ (2) ⇒ (3) ⇒ (1).

(1) ⇒ (2): Suppose that T has a perfect ranking, and consider vertices u, v, w with $u \to v$ and $u \to w$. We wish to show in this case that $u \to w$. Since the ranking is perfect, u appears before v and v appears before w, hence u appears before w. If we then had $w \to u$, this would be an upset, contradicting the fact that the ranking is perfect. Hence $u \to w$.

(2) ⇒ (3): Suppose that T is transitive. We wish to show that T contains no oriented cycle. We will assume the opposite, that T *does* have an oriented cycle. In this case, T must have an oriented cycle C of minimum length, say $v_1v_2v_3\cdots v_k$. But since T is transitive and since $v_1 \to v_2$ and $v_2 \to v_3$, we must have $v_1 \to v_3$. Thus $v_1v_3\cdots v_k$ is an oriented cycle of shorter length, contradicting the minimality of C. Hence T has no oriented cycles.

(3) ⇒ (1): First, suppose that T contains no oriented cycle. By the above lemma, T must contain a spanning path, say $v_1v_2\cdots v_n$. We claim that this path gives a perfect ranking. To see this, suppose that we have an upset, say $v_j \to v_i$ for $i < j$. In this case, we get an oriented cycle $v_iv_{i+1}\cdots v_{j-1}v_jv_i$, which is a contradiction. Hence the ranking has no upsets, and it is perfect. □

Okay, so we understand which graphs have perfect rankings. You will show on the homework that the probability that a random tournament on n vertices has a perfect ranking becomes vanishingly small as n gets large. So we must **abandon** our hopes of finding perfect rankings, because they **almost never exist!**

How can we save ourselves? What would be a more reasonable goal? Maybe we should look for rankings with a minimum number of upsets?

It turns out that this is a very difficult problem, and many people have suggested solutions, but it is difficult to say mathematically which method of ranking a tournament is best. It all depends what you mean by "best".

I will describe one method that I find to be very reasonable. If you have a round-robin ping-pong tournament this weekend, you are free to use this method.

The Kendall-Wei (1952,1955) method for ranking a tournament: Suppose that T is a tournament on vertices $\{v_1, \ldots, v_n\}$.

- Begin with the first strength vector $w_1 = (s_1(1), s_1(2), \ldots, s_1(n))$, where $s_1(i)$ is the number of vertices dominated by vertex v_i.
- Define the second strength vector $w_2 = (s_2(1), s_2(2), \ldots, s_2(n))$ by setting $s_2(i)$ equal to the sum of the *first strengths* $s_1(j)$ for vertices v_j dominated by v_i. (This takes account not only of how many people you defeated, but also how many people the people you defeated defeated!)
- Define the third strength vector $w_3 = (s_3(1), s_3(2), \ldots, s_3(n))$, where $s_3(i)$ is the sum of the *second strengths* of the vertices that v_i dominates.
- Continue in this way, to define the nth strength vector w_n. If n is quite large, we expect that w_n will give us an accurate assessment of the relative strengths of the players in the tournament.

However, computing w_n for large n might take a while. And there is the issue of "how large should we take n"? In practice, you might just go as far as you please. But it turns out that there is a precise way to calculate the **infinite limit** $\lim_{n\to\infty} w_n$ without too much trouble (although the proof is beyond the scope of this course):

Let M be the incidence matrix of the tournament T. That is, we let the (i,j)th entry of M equal 1 when $v_i \to v_j$, and let it equal 0 otherwise. Under certain fairly general conditions, M will always have a largest positive eigenvalue λ with 1-dimensional eigenspace. In this case the **ultimate strength vector** $w_\infty = \lim_{n\to\infty} w_n$ is proportional to any λ-eigenvector of M.

For example, the incidence matrix for the tournament above is

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

The largest positive eigenvalue of M is $\lambda \approx 1.395336994$ and the corresponding eigenvector (the ultimate strength vector) is

$$w_\infty \approx (.3213357548, .1650444157, .2833272505, .2302925789).$$

(You'll want to use a computer for this!) Thus, the Kendall-Wei method gives us the ranking (A, C, D, B). Do you think this is a reasonable answer? Let's just compare to what happens if we compute only the first few steps.

Consider again the tournament described above, and list the vertices in the order A, B, C, D, just for convenience. The first strength vector is

$$w_1 = (2, 1, 2, 1)$$

since A beats 2 players, B beats 1 player, C beats 2 players and D beats 1 player. This might not give a very "good" ranking since it has contains a lot of ties, and each defeated opponent contributes an equal amount to a player's "score". We would like to include information about "how strong were the opponents that you defeated?". So the second strength vector is

$$w_2 = (3, 1, 2, 2).$$

A gets *second strength* $3 = 1 + 2$ since A defeats B and C, who had *first strengths* 1 and 2, respectively. B has second strength 1 since B only defeated D, who has first strength 1. The second strength vector accounts for "how many opponents did the opponents that you defeated defeat?". To get the third strength vector, we sum the second strengths of defeated opponents.

$$w_3 = (3, 2, 3, 3)$$

And continue for a little while:

$$\begin{aligned}
w_4 &= (5, 3, 5, 3) \\
w_5 &= (8, 3, 6, 5) \\
w_6 &= (9, 5, 8, 8) \\
w_7 &= (13, 8, 13, 9) \\
w_8 &= (21, 9, 17, 13) \\
w_9 &= (26, 13, 22, 21) \\
w_{10} &= (35, 21, 43, 26).
\end{aligned}$$

How well does the tenth strength vector w_{10} compare to the ultimate strength vector w_∞? Since we only care about "relative strengths", we might as well rescale w_{10} so that the sum of its entries is 1 (as in "there is 1 unit of strength to go around, and we must decide who gets how much"):

$$\frac{w_{10}}{35 + 21 + 43 + 26} \approx (0.3017241379, .1810344828, .2931034483, .2241379310).$$

That's pretty darn close to w_∞ for only doing 10 steps, instead of infinitely many!