A Model for Intransitive Preferences

Sam Saarinen
University of Kentucky
Lexington, KY
samuel.saarinen@uky.edu

Craig Tovey
Georgia Tech
Atlanta, GA
ctovey@isye.gatech.edu

Judy Goldsmith
University of Kentucky
Lexington, KY
goldsmit@cs.uky.edu

Abstract

It is known that humans form non-transitive preferences. We propose to model human preferences as a complete weighted directed graph where nodes are objects of preference, and an edge leading from one node to another has weight equal to the probability that the destination node will be preferred (or chosen) from the pair connected by the edge. We also propose that the most preferred object(s) might be modeled as the one that wins in a round-robin tournament played amongst the nodes. We discuss computations using this model and introduce a polynomial-time algorithm to bound the probability of selecting an object from a pool of objects with symmetric preferences.

Introduction

We consider a probabilistic model of round-robin tournaments, or equivalently, Copeland voting, where the set of candidates equals the set of voters. We assume that the win probabilities for each game or pairwise vote are independent of all other win probabilities for other games. In particular, we do not assume that votes arise from voters' ranked orderings of candidates. We can treat such "games" as pairwise preferences, without assuming any form of transitivity.

Nontransitive Preferences

In 1969, Tversky pointed out that economic theories that depended on a notion of cardinal utility to define choices among alternatives assumed (and were equivalent to the assumption, when the set of alternatives was finite) that preferences were transitive (Tversky 1969). He presented arguments for the reasonableness of intransitive preferences, and more importantly, provided empirical evidence that Harvard undergraduates held intransitive preferences over gambles. He also discussed a stochastic model of preferences, where a is said to be preferred to b if the probability over time that a is preferred to b is $> 1/2$. He attributed the intransitive preferences to multi-criteria reasoning, where the evaluated criteria might vary from comparison to comparison.

Regenwetter and Dana offer another explanation of apparently intransitive preferences: that people may hold several rational preference orders, and choose among those orders (Regenwetter, Dana, and Davis-Stober 2011). The choice of order may depend on factors outside the model — when it's rainy, I prefer soup, when it's sunny, I prefer salad. They refer to "latent mental states" to describe internal factors. They found that the data collected from the subjects of Tversky's study and approximately 20 others were consistent with their preferences being a mixture (i.e., a probabilistic distribution) over linear preferences.

Probabilistic Tournaments

A round-robin tournament is one in which each team plays each other team. The winner is any team who wins the maximum number of games. This corresponds to a Copeland election in which the voters are exactly the candidates. Note that there may be a unique winner, or there may be a set of co-winners, teams who win the same maximum number of games.

Consider the setting in which the teams are known, and their pairwise probabilities of winning are also known. The problem of computing the probability of a given team winning the tournament was introduced by Mattei, et al. (Mattei, Goldsmith, and Klapper 2012), in the context of studying the complexity of computing winners and optimal bribery schemes for probabilistic models of elections (Binkele-Raible et al. 2013). They observed that the brute-force calculation of the probability of a team (say, University of Kentucky) winning a round-robin tournament involved evaluating the probabilities of every possible "scenario" of wins and losses, and adding up the probabilities of the scenarios in which the University of Kentucky wins at least as many games as any other team. This brute-force algorithm puts the calculation in the complexity class #P.[1] However, it is not clear that this problem is complete for #P, nor has it been proven hard for NP.

Consider a tournament in which A beats B and C but loses to D; B beats C and D, and C beats D. Then A and

[1]We can think of the class NP as languages corresponding to nondeterministic polynomial time Turing machines; for a given NP TM N, the language $\mathcal{L}(N)$ is the set of strings nondeterministically accepted by N. The corresponding #P function *counts* the number of nondeterministic computations of $N(x)$. It is known that $\#SAT$ is $\leq_m^P$-complete for #P, and for every L in the polynomial hierarchy, $L \leq_T^P \#SAT$.

B are *co-winners* of the tournament. In actual sports tournaments, there would be a play-off; in other settings, there may be another mechanism for determining a unique winner in such cases. For the purposes of this paper, we do not insist on unique winners.

Hazon, et al. (2008; 2012) considered a somewhat different probabilistic model of elections, closer to Regenwetter and Dana's model, in which voters have probability distributions over linear orderings of the candidates. In this setting, they were able to show that it is #P-complete to calculate the winner of a Copeland election. Crucially, another difference is that they did not restrict their attention to tournaments; the number of voters could be different from the number of teams/candidates. Thus, neither their result nor their proof has been useful for determining the complexity of our problem (evaluating the probability that a given team wins a round-robin tournament). However, like us, they considered Monte Carlo approximations of winner probabilities.

Michel Regenwetter has pointed out that we can view our model of probabilistic round robin tournaments as probability distributions over deterministic round robin tournaments, a.k.a., complete nontransitive preferences. Because we assume each game or match-up is independent of all others in the tournament, there is a one-one correspondence between determinizations of the edges and possible tournament graphs; the product of the probabilities associated with each deterministic edge is the probability of that deterministic tournament graph, given the probabilistic tournament graph that defines the probability distribution.

A New Model of Preferences

We model deterministic pairwise preferences as graphs, as follows. The objects in question are the nodes of the graph, and directed edges indicate the preference: the edge points to the more preferred object. We model probabilistic preferences as probability-weighted graphs, where a weight on a directed edge indicates the probability that the preference is aligned with that edge. Between each pair of connected nodes, there will be two edges, with the condition that the sum of the weights of these edges is 1, and each weight is nonnegative.

We assume all pair-wise match-ups have known probabilities (through either learning or elicitation). This will make our graph a complete directed graph. In this model of preferences, the most-preferred object corresponds to the most likely victor(s) of a round-robin tournament. Note that when n, the number of objects, is 2, this reduces to the known-probability case.

Computing Tournament Probabilities

Given the model introduced in the previous section, we become interested in a means of evaluating the probability of a given team winning a round-robin tournament.

Enumerative Solution

For small values of n, the likelihood that any given team is the winner of a round-robin tournament can be computed by enumeration.

Without loss of generality, index the teams from 1 to n. Initialize the probability that each team wins to 0. Iterate through all possible round-robin tournament outcomes. Compute each outcome's probability as the product of the probabilities of each game's outcome, since games are independent. Add the outcome's probability to the probability for each team that wins in that tournament outcome.

Even with constant-time multiplication at the requisite precision, a naive implementation of enumeration would take $\theta(n^2 2^{\binom{n}{2}})$ arithmetic operations, because each of the $2^{\binom{n}{2}}$ possible tournament outcomes has a probability equal to the product of $\binom{n}{2} = \theta(n^2)$ terms.

It is possible to reduce the naive time bound by enumerating the possible tournament outcomes with a Gray code, which is a sequence such that each outcome in the sequence differs from the next by reversing the outcome of one game. If a game outcome is reversed from i beats j to j beats i, then the probability of the next tournament outcome differs from that of the present outcome by a factor of $\frac{p_{j,i}}{p_{i,j}}$ (provided that each of $p_{i,j}$ and $p_{j,i}$ are nonzero), where $p_{i,j}$ is defined as the probability that i is preferred over j. Thus, except for the first ones, each probability and each Copeland score calculation takes only $O(1)$ arithmetic operations. However, updating team probabilities can still take $\Omega(n)$, giving an order n reduction in computation time. However, this does nothing to reduce the number of cases which must be enumerated.

The enumerative algorithm has the advantage of being exact (given sufficient numerical precision). Unfortunately, it is impractical for even very small tournament sizes (such as 10 teams).

Monte Carlo Evaluation

Another approach to evaluating the probability that a given team wins a round-robin tournament with probabilistic games is to randomly sample tournament events. For a fixed number of samples, or until some end condition is met, each game is randomly assigned a victor according to the probability that either team wins. The winners of the tournament event are noted, and each winner is credited with one 'sample'. After sampling has finished, the probability that each team wins the entire tournament is given by the 'samples' they won divided by the total number of sampled events.

This approach has the advantage of having controlled costs in terms of time, and statistically converging to the actual probabilities in the limit. However, it is also nondeterministic. In many high-stakes situations, it may be preferable to have an exact range, rather than to have a probability whose accuracy varies with the whim of a pseudorandom generator.

Further, there are cases when such a sampling approach will (for practical sample sizes) give an inaccurate expectation for tournament outcomes. For example, consider a tournament composed of 1 team with a very high probability of winning each of its games, and a large number of other teams with symmetric probabilities of beating each other (each has $\frac{1}{2}$ probability of beating the other in a match-up). In such a situation, there may be a non-negligible probability that some team other than the first team wins the entire tournament, and yet the probability of any one of the other teams winning is very small. In such a situation we could expect a Monte Carlo evaluation to return values which correctly predict that the first team may not win, but that do not reflect the fact that all of the remaining teams have equal probability among themselves (by symmetry) of winning the tournament.

Recursive Evaluation

The methods presented so far have been bottom-up, dealing with cases and then finding outcomes. Is it possible to approach this problem with a top-down method? That is, what if we consider outcomes, and then explore the relevant cases?

Let us make several observations. First, any proper subset S of the n teams may be thought of as playing a round-robin sub-tournament amongst themselves, after which the rest of the games in the full tournament are played. Second, if $|S| = n-1$, then the single team not in S, say t^*, has exactly 1 game with each of the elements in S, and no game with itself. Third, if t^* wins more games than the winner(s) of the sub-tournament, or wins as many games including all those against sub-tournament winners, then t^* is a winner of the entire tournament.

Denote the teams by the integers $1, \ldots, n$.

For $t^* \notin S \subset \{1, \ldots, n\}$, let $W(t^*, S)$ be the probability that t^* wins a round-robin tournament consisting of teams $t^* \bigcup S$.

Then $W(t^*, S) = \sum_{g=\lceil(|S|)/2\rceil}^{|S|} V(t^*, S, g)$ where $V(t^*, S, g)$ is the probability that team t^* wins in a tournament of teams $t^* \bigcup S$ and wins exactly g games. Let $B \subseteq S$ and $L = S \setminus B$ partition S. Define $P(t^*, B, L)$ to be the probability that team $t^* \notin S$ beats all of the teams in B and loses to all the teams in L. Then

$$P(t^*, B, L) = \prod_{k=1}^{|B|}(p_{t^*,b_k}) \cdot \prod_{k=1}^{|L|}(p_{l_k,t^*}).$$

Let $U(T, S^*, g^*)$ be the probability that the teams in set T are (all of the) co-winners of a tournament of teams $T \bigcup S^*$ ($T \cap S^* = \emptyset$) with each team in T winning exactly g^* games. Observe that

$$\begin{aligned} V(t^*, S, g) &= \sum_{B \subseteq S, |B|=g} P(t^*, B, S \setminus B) \\ &\cdot \left(\sum_{g_2=\lceil(|S|-1)/2\rceil}^{g-1} \sum_{T \subseteq S} U(T, S \setminus T, g_2) \right. \\ &+ \left. \sum_{T \subseteq B} U(T, S \setminus T, g) \right). \end{aligned}$$

Note that we deal separately with the cases when the winner of the sub-tournament had the same number of games as t^*.

In the case that $T = \{t_1\}$, $U(T, S, g) = V(t_1, S, g)$.

While these recursive definitions do not necessarily offer improvements on the other methods described in this section, they will form the basis for a polynomial-time algorithm in the next section which bounds the probability of tournament victory in the symmetric-probabilities case.

Case When All Probabilities are Equal

A Modified Recursive Algorithm

In the section on the recursive framework, we introduced a method of recursive computation of the probability of tournament victory which relied on a concept of sub-tournaments. In the general case, we provide no argument that this improves on the time complexity of an enumerative solution, but here we consider the special case that all the match-ups have symmetric probabilities. This will allow us to take advantage of problem symmetries to reduce our computational complexity.

Further, let us change our objective from computing the exact probability that a given team wins, and let us focus instead on placing upper and lower bounds on this probability.

Our solution will use a dynamic programming approach to short-cut redundant computations, and this will provide a means to apply an inductive argument to estimating the time complexity of this algorithm.

First, we define two tables, $T_C[|S|, g]$ and $T_E[|S|, g]$ which take the number of teams in the sub-tournament (the number of teams other than the one we are considering) and a number of victories and returns the lower and upper bounds on the probability that a given team (call it t^*, $t^* \notin S$) wins g games and is a co-winner or an exclusive winner, respectively, of the entire tournament. For the purpose of the following algorithms, let us say that values in these tables are initialized to some negative number.

We then define the overall probability of t^* winning in a symmetric-probabilities tournament with teams $t^* \bigcup S$ in terms of its probability of winning such a tournament with g victories, for each value of g (Algorithm 1).

Algorithm 1 Summing Across g.
Inputs: $|S|$. Outputs: An upper and lower probability

```
1: procedure PROBTOWIN
2:     for g ∈ {⌈|S|/2⌉ ... |S|} do
3:         sumUpper ← sumUpper + P_C(|S|, g).upper
4:         sumLower ← sumLower + P_C(|S|, g).lower
5:     return sumLower, sumUpper
```

To reduce the difference between the upper and lower bounds, we take the intersection of two separately computed ranges: one expresses the probability of winning; the other is the complement of the probability of losing. Note that in our algorithm we compute each probability of victory given that the winners get g victories, then we scale that by the probability that the winners do get g victories.

We attempt to reduce the difference between bounds by eliminating cases of overcounting and undercounting of ties (which are only estimated by our algorithm). In particular, we are interested in the winner of a subtournament, but we don't know who that may be; however, we may sum across the probability that each is the winner (nonexclusively and exclusively) to obtain upper and lower bounds. But these

Algorithm 2 Probability of t^* Being a Co-Winner.
Inputs: $|S|$, g. Outputs: An upper and lower probability

procedure P_C
  if $\neg(\lceil \frac{|S|}{2} \rceil \leq g \leq |S|)$ **then**
    return $0, 0$
  else if $|S| = g$ **then**
    return $\frac{1}{2^{|S|}}, \frac{1}{2^{|S|}}$
  else if $T_C[|S|, g] \geq 0$ **then**
    return $T_C[|S|, g]$
  else
    $sumUp1 \leftarrow 1 - P_{Losing}(|S|, g).lower$
    $sumLow1 \leftarrow 1 - P_{Losing}(|S|, g).upper$
    $sumUp2 \leftarrow P_{Winning}(|S|, g).upper$
    $sumLow2 \leftarrow P_{Winning}(|S|, g).lower$
    $sumUpper \leftarrow$ **min**$(sumUp1, sumUp2)$
    $sumLower \leftarrow$ **max**$(sumLow1, sumLow2)$
    $T_C[|S|, g] \leftarrow lower\frac{\binom{|S|}{g}}{2^{|S|}}, upper\frac{\binom{|S|}{g}}{2^{|S|}}$
    return $lower\frac{\binom{|S|}{g}}{2^{|S|}}, upper\frac{\binom{|S|}{g}}{2^{|S|}}$

Algorithm 3 Probability of t^* Being the Exclusive Winner.
Inputs: $|S|$, g. Outputs: An upper and lower probability

procedure P_E
  if $\neg(\lceil \frac{|S|}{2} \rceil \leq g \leq |S|)$ **then**
    return $0, 0$
  else if $|S| = g$ **then**
    return $\frac{1}{2^{|S|}}, \frac{1}{2^{|S|}}$
  else if $T_E[|S|, g] \geq 0$ **then**
    return $T_E[|S|, g]$
  else
    $sumUp1 \leftarrow 1 - P_{Losing}(|S|, g-1).lower$
    $sumLow1 \leftarrow 1 - P_{Losing}(|S|, g-1).upper$
    $sumUp2 \leftarrow P_{Winning}(|S|, g-1).upper$
    $sumLow2 \leftarrow P_{Winning}(|S|, g-1).lower$
    $upper \leftarrow$ **min**$(sumUp1, sumUp2)$
    $lower \leftarrow$ **max**$(sumLow1, sumLow2)$
    $T_E[|S|, g] \leftarrow lower\frac{\binom{|S|}{g}}{2^{|S|}}, upper\frac{\binom{|S|}{g}}{2^{|S|}}$
    return $lower\frac{\binom{|S|}{g}}{2^{|S|}}, upper\frac{\binom{|S|}{g}}{2^{|S|}}$

bounds can be improved. We may think of $P_C(|S|, g) - P_E(|S|, g)$ as the probability of there being a tie for the given team. Since all ties must involve at least two teams, and all teams' victories are represented, each tie must be counted at least twice, and we can subtract $\frac{|S|(P_C(|S|,g) - P_E(|S|,g))}{2}$ from the upper bound. Likewise, if we only consider the ties from one team, we know that none of these will be represented in the cases that each is an exclusive winner, so we may safely add $P_C(|S|, g) - P_E(|S|, g)$ to our lower bounds (Algorithms 4, 5).

Algorithm 4 Totalling the Probabilities that t^* Loses.
Inputs: $|S|$, g_{input}. Outputs: An upper and lower probability

procedure P_{Losing}
  $lower, upper \leftarrow 0$
  for $g \in \{g_{input} + 1 \ldots |S| - 1\}$ **do**
    $lower \leftarrow lower + P_C(|S|-1, g).lower + (|S|-1)P_E(|S|-1, g).lower$
    $over \leftarrow |S|\frac{P_C(|S|-1,g).upper + P_E(|S|-1,g).upper}{2}$
    $upper \leftarrow upper + over$
  end for
  $lower \leftarrow lower + P_C(|S|-1, g_{input}).lower + (|S|-1-g_{input})P_E(|S|-1, g_{input}).lower$
  $diff \leftarrow P_C(|S|-1, g_{input}).upper - P_E(|S|-1, g_{input}).upper$
  $fracTies \leftarrow 1 - \frac{|s|-1-g_{input}}{2(|s|-1)}$
  $ties \leftarrow diff \cdot fracTies$
  $upper \leftarrow upper + (|S| - g_{input})(P_E(|S|-1, g_{input}).upper + ties)$
  return lower,upper

Algorithm 5 Totalling the Probabilities that t^* Wins.
Inputs: $|S|$, g_{input}. Outputs: An upper and lower probability

procedure $P_{Winning}$
  $lower, upper \leftarrow 0$
  for $g \in \{0 \ldots g_{input} - 1\}$ **do**
    $lower \leftarrow lower + P_C(|S|-1, g).lower + (|S|-1)P_E(|S|-1, g).lower$
    $over \leftarrow |S|\frac{P_C(|S|-1,g).upper + P_E(|S|-1,g).upper}{2}$
    $upper \leftarrow upper + over$
  end for
  $lower \leftarrow lower + g_{input}P_E(|S|-1, g_{input}).lower$
  $diff \leftarrow P_C(|S|-1, g_{input}).upper - P_E(|S|-1, g_{input}).upper$
  $fracTies \leftarrow 1 - \frac{g_{input}-1}{2(|s|-1)}$
  $ties \leftarrow diff \cdot fracTies$
  $upper \leftarrow upper + g_{input}(P_E(|S|-1, g_{input}).upper + ties)$
  return lower,upper

Time Complexity and Limitations of the Equal-Probabilities Algorithm

We may estimate the time complexity of this algorithm by considering a bottom-up approach to filling our dynamic

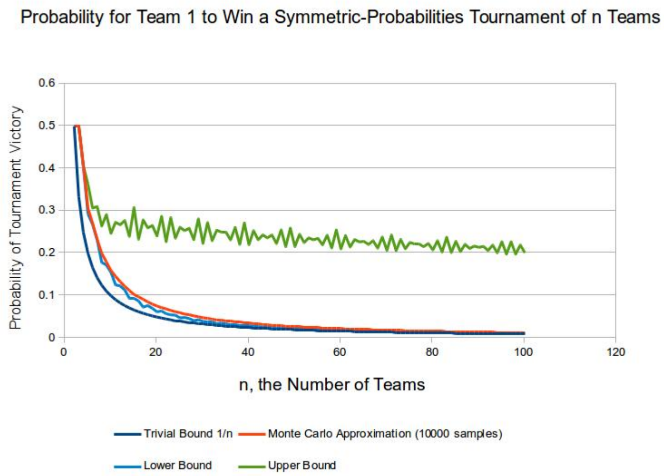

Figure 1: Empirical Examination of the Bounds and Expected Values for Victory in a Symmetric-Probabilities Tournament. Note the improvement of our lower bound over the trivial lower bound: $\frac{1}{n}$.

programming tables (which we assume has $O(1)$ access time). For a number of teams in the tournament n, we will need to consider sub-tournaments with $\{1, 2, 3, ..., n-1\}$ teams. In each of these, we will be interested in cases when the winner of the sub-tournament has $\{0, 1, 2, ..., n-2\}$ wins. This means that the number of entries in our table will be $O(n^2)$. Further more, filling a new entry in the table uses constant-time operations on at most n other table entries (since we iterate over number of wins for a constant number of teams in the sub-tournament). This makes the time to fill the tables $O(n^3)$. Since computing the total probability of a team winning the entire tournament requires a sum across at most n elements in the table, this will be dominated by the cost of filling the table. This polynomial-time algorithm makes it computationally feasible to evaluate symmetric-probabilities tournaments with an order of magnitude larger number of teams than is feasible with the enumerative solution.

Note that the upper bound can be shown empirically to be too weak for accurate prediction of the behavior of large tournaments (See Figure 1). However, we have no evidence that these are the strongest bounds obtainable with this time complexity. An area of future research may be the strengthening of these upper and lower bounds. It is possible that using related measures of the likelihood of the existence of a unique winner or enumerating the number of possible winners will provide stronger bounds with minimal computational overhead.

With improved bounds, or used with cases with a small number of teams, an algorithm such as this one can be coupled with the recursive framework described earlier to efficiently evaluate or approximate a tournament over a subset of teams in the general case where that subset has the special property of containing only symmetric probabilities. This technique could also be applied to deterministic subsets of the tournament (where match-ups have probabilities 1 and 0 respectively). The results from this efficiently computed sub-tournament may then be used in the evaluation of the remainder of the tournament.

Finally, it may be possible to generalize this technique of bounding the probability of tournament victory to a more general case by identifying subsets of the tournament-outcome space which are expensive to enumerate, but where probability of each team's victory across this subset can be cheaply bounded.

Conclusions

In this paper we have introduced a new model of preferences, we have described a recursive method of evaluating a tournament which can take advantage of simpler substructures in a larger tournament, and we have introduced a polynomial time algorithm to bound the probability of victory in an symmetric-probabilities round-robin tournament. The model we introduced suggests that we may be able to model other probabilistic (potentially non-transitive) dominance relationships using preferences. For example, we may model dilemmas from game theory or even military combat using preferences. Thus, being able to reason about nontransitive preferences may have not only benefits for our personal lives (when gambling on sports tournaments, for example), but also a profound societal influence through economics, psychology, and perhaps even military strategy.

Acknowledgments

This work is partially supported by the National Science Foundation, under grant CCF-1215985. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

We thank Martin Mundhenk for pointing out that the equal-probabilities case should be at least as easy as the general case of computing the probability of a team winning a round-robin tournament.

References

Binkele-Raible, D.; Erdélyi, G.; Fernau, H.; Goldsmith, J.; Mattei, N.; and Rothe, J. 2013. The complexity of probabilistic lobbying. *Discrete Optimization*.

Hazon, N.; Aumann, Y.; Kraus, S.; and Wooldridge, M. 2008. Evaluation of election outcomes under uncertainty. In *Proc. AAMAS*.

Hazon, N.; Aumann, Y.; Kraus, S.; and Wooldridge, M. 2012. On the evaluation of election outcomes under uncertainty. *Artificial Intelligence* 189:1–18.

Mattei, N.; Goldsmith, J.; and Klapper, A. 2012. On the complexity of bribery and manipulation in tournaments with uncertain information. In *Proc. FLAIRS*.

Regenwetter, M.; Dana, J.; and Davis-Stober, C. P. 2011. Transitivity of preferences. *Psychological Review* 118(1):42.

Tversky, A. 1969. Intransitivity of preferences. *Psychological review* 76(1):31.