CS599: Structure and Dynamics of Networked Information (Spring 2005)
02/23/2005: Rank Aggregation
Scribes: Ranjit Raveendran and Animesh Pathak

In previous lectures, we had discussed the problem of searching for relevant results on the WWW by exploiting the link structure. Nowadays, there are already multiple search engines giving quite good results. However, given that there are so many search engines already, employing different techniques, we may be interested in combining their positive features, and constructing a meta-search engine that uses all of their results. This leads very naturally to the problem of *rank aggregation*: given several orders on items (such as search results), determine a *consensus ordering*, which somehow reflects all orderings together.

This problem can be studied from different perspectives. We could treat it as a machine learning problem, by treating each search engine as an "expert", and trying to learn which engine's advice to trust, based on past performance. The goal would then be to perform no worse than the best expert, but without knowing ahead of time which one is actually the best expert.

Another approach, which we will explore in future lectures, is to consider it as an optimization problem. By defining an appropriate notion of distance between rankings, we can then look for a ranking that is close to all given rankings in a certain sense.

A third approach, to be pursued in this and the following lecture, exploits the close connection between rank aggregation and voting. In both settings, we have items (candidates or web pages), and orderings on them (voter preferences or search engine rankings). The goal is to find a ranking that is "agreeable" to all voters/search engines. By using this analogy, we can leverage several centuries of thought on the issue of voting and social choice.

1 Rank Aggregation as Social Choice

To get a feel for the difficulties in determining a consensus ordering, let us look at a simple example where a first choice is to be determined. Suppose there are three candidates, GWB, AG, and RN, and the voters' preferences are as follows:

- 49% of voters prefer the order GWB–AG–RN
- 48% of voters prefer AG–GWB–RN
- 3% of voters prefer RN–AG–GWB

Who should be considered the winner legitimately? There are different plausible answers, depending on the view we take of these preferences. We could argue that GWB was the candidate desired as winner by the largest number of voters, and hence should be the winner. This is the outcome of plurality voting. On the other hand, we could argue that a majority of voters prefers AG over GWB, and also over RN. Hence, AG wins all pairwise comparisons, and should be the winner.

One of the earliest approaches to solving this problem is due to Borda (1770). His method essentially uses the average position of a candidate, averaged over all voters' preferences, and sorts candidates in this order. More formally, for each voter, each candidate obtains a score equal to the number of other candidates he beat in this voter's ranking. Candidates are then ranked by the sum of their scores, summed over all voters. In our above example, this would give us AG as the winner, as the score would be $49 \cdot 1 + 48 \cdot 2 + 3 \cdot 1 = 148$, whereas the score of GWB is $49 \cdot 2 + 48 \cdot 1 = 146$ (and the score of RN is $3 \cdot 2 = 6$).

Although Borda's rule looks useful at first, a major problem with it is that even the candidate with the largest number of pairwise wins can lose. For instance, if 51% of the voters prefer the order ABC, and 49%

prefer BCA, A's score is $51 \cdot 2 = 110$, while B scores $49 \cdot 2 + 51 = 149$, thus winning the election though clearly more people prefer A to B than vice versa.

Taking the latter idea further, we may wish to attain the following property, called *Condorcet Criterion*: If candidate A beats candidate B in pairwise comparison (i.e., more voters prefer A to B than vice versa), then A should precede B. We quickly run into a problem. If three voters have the preference orders ABC, BCA, and CAB on three candidates A,B,C, then A should precede B, B should precede C, and C should precede A. Obviously, not all three can be accomplished simultaneously.

A relaxed version of the property, called *Extended Condorcet Criterion (ECC)* (or *General Condorcet Criterion (GCC)*) is as follows: If X, Y are a partition of the set V of all candidates (i.e., $X \cap Y = \emptyset$ and $X \cup Y = V$), and for all $x \in X, y \in Y$, x beats y in direct comparison, then all of X should precede all of Y.

This version is much less restrictive; in particular, for the example given above, it allows us to choose an arbitrary ranking of candidates. In fact, we can show that for any input orderings, there is always an ordering satisfying the GCC.

Proposition 1 *For any input rankings, there is a consensus ordering satisfying the GCC.*

Proof. Consider the directed graph G on V with an edge from x to y if x beats y in pairwise comparison (we also write $x > y$ for this). Notice that this is a *tournament graph*, i.e., a graph in which for each pair of nodes, there is an edge one way or the other. We prove below that every tournament graph contains a Hamiltonian Path. Consider the candidate ordering determined by the sequence of vertices along such a Hamiltonian Path. Assume that this ordering fails the GCC. Then, there is a partition (X, Y) of V such that each $x \in X$ beats each $y \in Y$ in direct comparison, yet some $y \in Y$ precedes some $x \in X$ in this ordering. Then, there must also be an *adjacent* such pair in the ordering, i.e., one where y immediately precedes x on the Hamiltonian Path. But this is a contradiction, as there must have been an edge from y to x in the Hamiltonian Path, so y must actually beat x in direct comparison. ■

Lemma 2 *Each tournament graph G contains a Hamiltonian Path.*

Proof. We use induction on the number of vertices, n. For $n = 1$, the claim is trivial. For $n \geq 2$, let $x \in V$ be an arbitrary vertex. By induction hypothesis, $G[V - x]$ has a Hamiltonian path $x_1, x_2, \ldots, x_{n-1}$. If there is an edge from x to x_1, or from x_{n-1} to x, then insert x at the beginning or end of the ordering, respectively. Otherwise, there must be a k with $1 \leq k \leq n - 1$, such that there is an edge from x_k to x, and from x to x_{k+1} (start from x_1, and follow the path until a node has an edge from x). By inserting x between x_k and x_{k+1}, we obtain a Hamiltonian path for G. ■

2 Formalization of Social Choice Properties

So far, we have investigated several approaches for determining a consensus ordering from several given orderings. All approaches suffered from "unnatural" outcomes in some cases or others. Perhaps, it would thus make sense to axiomatize which properties a voting scheme should satisfy, and then look for voting schemes meeting these axioms. We describe here the axioms proposed by Arrow [1]

Formally, we will associate with each voter i a preference order $\prec_i$. A *social choice function* f takes all of these k orders, and outputs a consensus order

$$\prec \quad = \quad f(\prec_1, \ldots, \prec_k).$$

Such a function f should intuitively satisfy certain properties.

Monotonicity : If $a \prec b$, and $b \prec_i a$, then swapping a and b in $\prec_i$ does not result in $b \prec a$. Intuitively, this means that if a ranks above b overall, then changing another vote in a's favor does not affect the relative ranking between a and b.

Non-triviality : For each pair a and b of candidates, there is some choice of orderings $\prec_i$ such that $a \prec b$. This ensures that the relative ordering of a and b actually depends on the votes, and is not predetermined.

Independence of Irrelevant Alternatives (IIA) : Let $\prec_1, \ldots, \prec_k$ and $\prec'_1, \ldots, \prec'_k$ be two different preference orders for each voter, and $\prec = f(\prec_1, \ldots, \prec_k)$ and $\prec' = f(\prec'_1, \ldots, \prec'_k)$ the corresponding consensus orderings. If $B \subseteq V$ is a subset of candidates such that $a \prec_i b$ if and only if $a \prec'_i b$ for all $a, b \in B$ (i.e., all $\prec_i$ and the corresponding $\prec'_i$ agree on the orderings of B), then $a \prec b$ if and only if $a \prec' b$ for all $a, b \in B$. What this expresses is that if no voter changes his relative preference between any two candidates in B, then the final ordering among candidates of B does not change. In other words, changing preferences merely with regards to third candidates does not affect the order of any two other candidates.

Monotonicity and non-triviality together imply the property of *unanimity*: if $a \prec_i b$ for all i, then $a \prec b$. That is, if every voter prefers a over b, then a ends up ahead of b. To prove this fact, start with some set of orders $\prec_i$ such that $a \prec b$ (such a set exists by the non-triviality property). Then, we keep swapping the positions of a and b in all $\prec_i$ that previously had $b \prec_i a$. By monotonicity, the outcome will still be that a is ranked ahead of b, and eventually, $a \prec_i b$ for all i.

In trying to find social choice functions satisfying all these axioms, one quickly notices that this is not so easy. In particular, the IIA property is not satisfied by many schemes. However, one class of social choice functions meeting these requirements is *dictatorship*: the dictator function f_i is defined as $f_i(\prec_1, \ldots, \prec_k) = \prec_i$. That is, the output is simply the preference of just *one* voter. Dictatorship is an undesirable quality for a voting scheme. That gives us our last property:

Non-Dictatorship : $f \neq f_i$ for all i. That is, the aggregation will not disregard the opinions of all but one voter.

Unluckily, with this additional requirement, we have ruled out all remaining social choice functions:

Theorem 3 (Arrow, 1951 [1]) *There is no function f satisfying all the above four properties. Hence, the only functions satisfying the first three properties are the dictatorship functions.*

References

[1] K. Arrow. *Social Choice and Individual Values.* Wiley, 1951.