PROBLEMS WITH VOTING METHODS

The problems with voting methods involving three or more candidates are always discussed in terms of the raw data--the votes--and how various arrangements of these data can produce illogical results. For example, an arrangement is said to violate *consistency* if a winner in two separate sub-elections can become a non-winner when the votes are combined into a single election. The arrangement violates *monotonicity* if an increase in the votes given to the winner changes his winning status (or a decrease applied to a non-winner changes him to a winner). *Violation of independence from irrelevant alternatives* occurs when votes shifted among low-level candidates cause outcome changes among high-level candidates. Yet another requirement involves the *Condorcet criterion*: If a candidate beats all others in pairwise contests, he ought to win the election. This last point requires a bit of explanation.

To illustrate the Condorcet criterion, consider that candidate A receives various preference rankings by the voters. Other candidates receive different rankings by the same voters. Now ask the following question: For how many of the voters did candidate A receive a higher ranking than candidate B (or C, or D, etc.)? These numbers (A vs B, A vs C, A vs D, etc.) can form the 'A' row of a candidate-vs-candidate matrix, and the process can then be repeated for candidate B, forming the second row of the matrix. When the matrix is complete, the row sums can be compared, and the candidate whose row sum is greatest is called the Condorcet winner. He has beaten the other candidates in the judgment of the most voters. The logical expectation is that the winner in any vote-processing scheme should also be the Condorcet winner. Condorcet gives frequent ties, so we say that a voting method fails the Condorcet test if its winner is not at least tied for first place in a Condorcet computation.

In all of these tests of logical behavior, developed over centuries, the vote data are being treated as scalar quantities. But the mechanical analog of an election uses votes as *vector* expressions of preference, and preferences go both ways--from voter to candidate and from candidate to voter. To make the tests usable for the vector approach, the vote vectors must be converted to scalars, requiring a scalar product involving another vector. The obvious choice for this other vector in the scalar product is the *contest vector* (the vector sum of all of the vectors in the election).

The raw data in an election are the votes for the candidates, expressed as normalized preference distribution functions. The candidates reflect the votes they receive back to the voters who sent them, thus expressing preferences for the voters. The strength of a participant (voter or candidate) is called X_i and the j component of the strength vector is called $\underline{X}_{ij}$, where the index j is for the direction of the other participant toward which the vector is pointing. The equations of the contest algorithm show how the X quantities depend upon each other. These equations are written and discussed in many places on this web site, but I will write the equation for X_i again here, although it is not needed for the present discussion:

$$X_i^2 = \Sigma_j \, [A_{ij} \cdot f_{ij} \cdot X_j] \quad \text{where } f_{ij} = A_{ij} / \Sigma_i \, (A_{ij})$$

The quantities A_{ij} are the vote preferences: participant i for participant j and the other way around, both numbers the same. The square root of the bracketed quantity is the direction-j component of participant i's strength vector.

The sum of all of the strength-vector projections (both sides) upon the contest vector will be equal to the magnitude of the contest vector, and we now have a scalar problem with vector projections taking the place of the votes in the original problem. But the projections involve the strengths of all of the participants in the election, since all strengths become interconnected in the contest algorithm which computes them. Any change in the input data will ripple through the whole contest and produce some change in every participant's strength. This means that any consideration of what happens when the election is divided into sub-elections must come *after* the strength determinations for the election as a whole. For the *consistency* test, described above, there can be no such thing as two separate elections, a winner in both, and a test to see if the winner in both is still the winner when the votes are combined into one election. This is because the case of two separate elections constitutes a different problem: Many of the interactions among participants have been removed in the two-election case. Looking at it from the perspective of the mechanical model used for deriving the set of equations, the participants must apply brakes simultaneously against different numbers of opposing participants in the two situations, and this affects their computed strengths.

A consistency requirement modified to suit the vote-vector model would be the statement that any *post-analysis* division of the full electorate into portions, with a particular candidate winning in every portion, must have that candidate winning in the full election. The proof that this must be true for every election is trivial, since a candidate having the largest vector projection in every portion must have the largest sum (over all portions) of vector projections.

The *monotonicity* requirement says that a winner must remain a winner if any voters change their preferences downward for any candidates except that winner. Because of the normalization of preferences which the vector method uses, revisions of voters' preferences downward must be accompanied by upward changes of preference for the winner. One of the main features of the contest algorithm is monotonicity, in the sense of increasing *strengths* of participants going along with increases of appropriate scores in the scoreboard. The monotonicity requirement of an *election*, in the context of the vector method, calls for an increase in the strength of a winning candidate when scoreboard changes are in his favor--a somewhat different statement from monotonicity applied to strengths. Nevertheless, I am quite confident that the vector method satisfies election monotonicity in every instance. I do not have a proof for this, just a conjecture, but help with that conjecture comes from the double effect which normalization produces: increasing the strength for the winner and decreasing the strengths of non-winners.

The fairness criterion called *independence from irrelevant alternatives* (or *independence* for short) is violated by the Borda count--a way of processing votes favored because of its simplicity. With Borda, the candidates are given weights (typically integers) to go along with their priority standings, and the weights are added up for each candidate. It can allow order-of-preference changes by voters among the low-preference candidates to affect the choice of winning candidate. This criterion cannot apply to election processing by contest analysis because of the system of interlocking vectors which connects all participants--a change in one place affecting the whole system. Thus, a low-level candidate can have just enough effect on a close pair of high-level candidates to determine the winner. Indeed, we look to the low-level candidates in many cases to give the extra discrimination needed for breaking ties. Independence is not a criterion of fairness when the vector method is used, and no alternatives are irrelevant.

The only major test of reasonableness for an election yet to be dispensed with is the *Condorcet criterion*--either by establishing that the vector method satisfies it or that the test really should not apply when interactions are present. Condorcet considers preference rankings, but it can be modified to take account of the more-precise *cardinal* preferences given by the vote-vector method. Also, the voters now have strength differences, so it can no longer be argued that Condorcet is an expression of the equal treatment of all voters. Even though all voters cast an equal number of votes and appear equal at the ballot box, the pattern of choices will determine how strongly each voter influences the election. The Condorcet winner and the Borda winner can be computed from the same matrix of vector projections and it would be nice if the two methods were to always select the same winner. The following paragraph will prove that this is the case.

Proof that the Condorcet method, applied to vote-vector data, will reproduce the vote-vector ordinal ranks:

For this proof, I will avoid the notational complexity associated with the vote vectors and merely state that the strength projection for candidate i has length L_i and that the sum of all projections for the candidates is unity. Now each candidate's length segment is divided into sub-segments of varying lengths, labeled l_{ij}. The index j is for the voters, and the sub-segments represent the contributions to the candidate's strength from the voter groups. Next compute the difference in sub-segment length for voter j between two candidates (i and k), and then summed over all voters. Calling this sum of differences d_{ik}, we have

$$d_{ik} = \Sigma_j \, (l_{ij} - l_{kj}).$$

This is what Condorcet is doing when it finds the number of voters preferring candidate i over k, but here we are using more precise preference numbers and letting the individual differences be both positive and negative. (This is an improvement over the original Condorcet, where the d values are restricted to 1 and 0 for each voter, representing 'yes' and 'no' for preference. I call this the *revised Condorcet* method.) Next, sum over k to find the overall difference sum between candidate i and all of the other candidates. The k sum must exclude i, so n candidates will give n–1 terms in the summation. If the overall sum of differences is D_i , then

$$D_i = \Sigma_{k\neq i} (d_{ik}) = (n-1) \cdot \Sigma_j \, (l_{ij}) \; - \; \Sigma_{k\neq i,j} (l_{kj}) = (n-1) \cdot L_i \; - \; \Sigma_{k\neq i} (L_k)$$
$$= (n-1) \cdot L_i \; - \; (1 - L_i) \; = \; n \cdot L_i - 1.$$

The Condorcet winner is the candidate with the largest value of D and that winner also has the largest value of L. The electorate's preference order for the other candidates will go along with their D values; the lowest ranking one will have the most negative value of D. The L values are actual lengths of projections and always positive, so they are used to determine the relative numerical standings of the candidates. Not only is there agreement between Condorcet and the vector method on the winner, when both methods operate on the same data, but the agreement extends to the order of placement of every lesser candidate.

If Borda counts had been used for the data instead of strength-vector projections, a similar calculation would show ordinal ranks equal for Borda and revised Condorcet.

Now that it has been shown that Borda, Condorcet (revised), and vector methods all give the same ordinal ranks on candidates when they operate on vector projections, there is no need to distinguish between Borda and Condorcet--in fact, no need to keep them around any longer. They cannot be properly used unless the vector method gives the strengths, and having the strengths renders the problem solved.

A summary of what I have presented above about the four fairness criteria and how the vector method treats them is listed here:

Consistency ----- Never violated (post analysis)
Monotonicity ----- Never violated (conjecture)
Independence ----- Explained away
Condorcet criterion ----- Explained away

A remark should be made about *strategic voting*, or voting to obstruct an unfavored candidate and not to express true preferences. The extent to which vote-aggregation methods encourage strategic voting is sometimes used to pick one aggregation method over another, recognizing that strategic voting is a natural inclination among voters. Methods which make it hard for the voter to devise an obstructive vote will encourage the voter to express his true preferences. Complexity in the vote-processing method will discourage strategic voting, and the vector method is certainly highly complex. It gets nowhere without a computer program, and the voter determined to use a strategy would have to run "what if" examples to decide upon his strategy. In so doing, he might find that an expression of preference for his minor candidate can have an effect on the major candidates and that he might do nearly as much good with an "honest" vote as with a strategic one. In any case, it does not make sense to choose an incorrect election method because it might be more confusing to the voter and therefore less open to strategic voting.

--- Alan E. Johnsrud

24 January 2000 (last revision 28 October '09)

Go to Home page: [Home] Go to Topics page: [Topics]