Claremont Colleges
Scholarship @ Claremont

All HMC Faculty Publications and Research   HMC Faculty Scholarship

9-1-2008

Borda Meets Pascal

Marie K. Jameson
University of Wisconsin - Milwaukee

Gregory Minton '08
Harvey Mudd College

Michael E. Orrison
Harvey Mudd College

Recommended Citation

Borda Meets Pascal

Marie K. Jameson, Gregory Minton, and Michael E. Orrison
University of Wisconsin and Harvey Mudd College

Every so often (especially in mathematics), unforeseen connections between different ideas arise and beg explanation. This happened to us when, in an effort to generalize the voting procedure known as the Borda count, we began to see vectors of the form $(-1,1), (1,-2,1), (-1,3,-3,1)$, $(1, -4, 6, -4, 1)$, and so on. As you might imagine, we were instantly intrigued by this unanticipated relationship with Pascal's triangle, and we quickly set out to find an explanation. This article describes some of our initial findings.

The Borda Count

The Borda count is an extremely well-known and well-studied voting procedure. It works by first asking each voter to rank the candidates in an election from most preferred to least preferred, i.e., from first to last. Then, letting n be the number of candidates, each candidate is given $n - 1$ points for each first place vote, $n - 2$ points for each second place vote, $n - 3$ points for each third place vote, and so on. The candidate who receives the most points is then declared the winner.

The Borda count is a specific example of a *positional voting* procedure, where candidates receive points based on the position they occupy on a voter's ballot. Each such procedure for n candidates is defined by its *weighting vector* $w = (w_1, ..., w_n)$, where being placed in position i means that a candidate receives w_i points. For example, the weighting vector associated with the Borda count is $(n - 1, n - 2, ..., 1, 0)$.

A particularly attractive feature of positional voting is that results may be realized as matrix-vector products. For example, suppose there are three candidates A, B, and C. Furthermore, suppose there are 15 voters with the following preferences: 3 voters prefer A to B to C; 4 voters prefer A to C to B; 2 voters prefer B to C to A; and 6 voters prefer C to B to A. We can encode this information in a *profile* vector

$$\mathbf{p} = \begin{bmatrix} 3 \\ 4 \\ 0 \\ 2 \\ 0 \\ 6 \end{bmatrix} \begin{matrix} ABC \\ ACB \\ BAC \\ BCA \\ CAB \\ CBA \end{matrix}$$

where, for ease of understanding, we have written the corresponding preferences which index the entries to the right. In this case, the result of the election with respect to the weighting vector (w_1, w_2, w_3) is given by the product

$$\begin{bmatrix} w_1 & w_1 & w_2 & w_3 & w_2 & w_3 \\ w_2 & w_3 & w_1 & w_1 & w_3 & w_2 \\ w_3 & w_2 & w_3 & w_2 & w_1 & w_1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 0 \\ 2 \\ 0 \\ 6 \end{bmatrix} \begin{matrix} ABC \\ ACB \\ BAC \\ BCA \\ CAB \\ CBA \end{matrix} = \begin{bmatrix} 7w_1 + 8w_3 \\ 2w_1 + 9w_2 + 4w_3 \\ 6w_1 + 6w_2 + 3w_3 \end{bmatrix} \begin{matrix} A \\ B. \\ C \end{matrix}$$

For example, if we use the Borda count where $(w_1, w_2, w_3) = (2, 1, 0)$, then the result would be $(14, 13, 18)$, meaning that candidate C wins. On the other hand, if we use the weighting vector $(1, 0, 0)$ (i.e., everyone is asked to "vote for your favorite"), then the result would be $(7, 2, 6)$, meaning that candidate A wins. As these simple examples might suggest, being able to explain how and why different voting procedures can lead to different outcomes for a fixed set of preferences is one of the reasons voting theory is such a fascinating subject.

Why is the Borda Count Special?

The Borda count has many interesting properties. The one we will focus on here is the fact that the results of a Borda count election can be recovered from the results of the so-called *pairwise map*. This map counts the number of times that each candidate beats each other candidate. For example, with three candidates A, B, and C, it tells us how many times A was ranked above B, B was ranked above A, A was ranked above C, and so on.

Once again, all of this can be encoded as a matrix-vector product. For example, if we consider the election with 15 voters described above, the result of the pairwise map is the product

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 0 \\ 2 \\ 0 \\ 6 \end{bmatrix} \begin{matrix} ABC \\ ACB \\ BAC \\ BCA \\ CAB \\ CBA \end{matrix} = \begin{bmatrix} 7 \\ 7 \\ 8 \\ 5 \\ 8 \\ 10 \end{bmatrix} \begin{matrix} AB \\ AC \\ BA \\ BC \\ CA \\ CB \end{matrix}.$$

To see the connection to the Borda count, note that the score awarded to each candidate by the Borda count election can be recovered from the result of the pairwise map by simply adding the entries in which that candidate beat another candidate. For example, A beat B 7 times, and A beat C 7 times, so A's Borda count score is 14. This makes sense, though! After all, if A is ranked first out of 3 candidates, then A will beat B and A will beat C. In other words, A wins 2 head-to-head competitions. If A is ranked in the middle between B and C, then A will only win 1 head-to-head competition, and if A is ranked last, then she will win 0 head-to-head competitions. Hence we get the weighting vector $(2, 1, 0)$!

In general, we may think of the Borda count as aggregating pairwise information, and its associated weighting vector $(n-1, n-2, ..., 1, 0)$ as corresponding to a simple counting question: If a candidate is ranked by a voter in the ith position, then how many other candidates will she be ranked above? The answer, of course, is $i - 1$, which leads us to the weighting vector associated with the Borda count.

Before we say any more about the Borda count, it will be helpful to introduce some notation and a definition. We denote the pairwise map by $P_2 : R^{n!} \rightarrow R^{n(n-1)}$. Next, let U, V, and W be real vector spaces, and let $T : U \rightarrow V$ and $T' : U \rightarrow W$ be linear transformations. We say that T' is *recoverable* from T if there exists a linear transformation $R : V \rightarrow W$ such that $T' = R \circ T$. As an aside, this is equivalent to saying that the kernel of T is a subset of the kernal of T', which can be useful when actually trying to determine when one linear transformation is recoverable from another.

Suppose there are n candidates in an election, and that $\mathbf{w} = (w_1, ..., w_n) \in R^n$ is a weighting vector. We will denote the linear transformation corresponding to the positional voting procedure for $\mathbf{w}$ by $T_w : R^{n!} \rightarrow R^n$. Let $\mathbf{b}_1 = (1, ..., 1) \in R^n$ be the all-ones vector, and let $\mathbf{b}_2 = (n-1, n-2, ..., 1, 0) \in R^n$ be the weighting vector for the Borda count. The following theorem is a nice example of why the Borda count is so special:

Theorem 1. Let $n \geq 2$ and let $\mathbf{w} \in R^n$ be a weighting vector. The map T_w is recoverable from the pairwise map P_2 if and only if $\mathbf{w}$ is a linear combination of $\mathbf{b}_1$ and $\mathbf{b}_2$.

This theorem is particularly interesting to voting theorists because there are many voting procedures (e.g., the well-known Copeland method) that are essentially based on the results of the pairwise map P_2. This theorem may therefore be seen as a first step to realizing that, unless your positional voting procedure is essentially the Borda count (since $\mathbf{b}_1$ only contributes to ties and will therefore never distinguish one candidate from another), you should not be surprised if the outcome disagrees with that of any voting procedure that relies solely on pairwise information.

Generalizing the Borda Count

Viewing the Borda count in light of Theorem 1 immediately suggests a possible generalization. Instead of focusing on information about pairs, what if we considered information about triples, quadruples, and so on? Since the Borda count is related to pairs information by counting the number of times a candidate was ranked above some other candidate, we might look at candidates being ranked above subsets of other candidates.

For example, we could ask: How many times was each candidate ranked above a $(k - 1)$-element subset of other candidates? This question can be answered easily by using the weighting vector

$$\mathbf{b}_k = \left(\binom{n-1}{k-1}, \binom{n-2}{k-1}, ..., \binom{1}{k-1}, \binom{0}{k-1} \right)$$

since it would assign points to a candidate based on the number of $(k - 1)$-sized subsets of candidates ranked below her. For example, if $n = 5$, then

$\mathbf{b}_1 = (1, 1, 1, 1, 1)$, $\mathbf{b}_2 = (4, 3, 2, 1, 0)$, $\mathbf{b}_3 = (6, 3, 1, 0, 0)$,

$\mathbf{b}_4 = (4, 1, 0, 0, 0)$, $\mathbf{b}_5 = (1, 0, 0, 0, 0)$.

We can also generalize the pairwise map P_2 to create the k-wise map $P_k : R^{n!} \rightarrow R^{(n)_k}$ where

$$(n)_k = n(n-1)(n-2) \cdots (n-k+1)$$

is "n falling factorial k." The map P_k counts the number of times each ordered k-tuple of candidates is actually ranked in that order by a voter. For example, if a voter chooses the order $ABCD$ for four candidates, and $k = 3$, then this voter would contribute to the totals for the following ordered 3-sets: ABC, ABD, ACD, and BCD.

Note that P_1 just gives each candidate a score that is equal to the total number of voters, and that P_n is the identity map. Furthermore, it is easy to see that $T_{\mathbf{b}_k}$ is recoverable from P_k, and that P_j is recoverable from P_k whenever $j \leq k$. These facts, together with some technicalities (which are slightly beyond the scope of this article), lead to the following generalization of Theorem 1:

Theorem 2. Let $n \geq 2$ and let $\mathbf{w} \in R^n$ be a weighting vector. The map $T_\mathbf{w}$ is recoverable from the k-wise map P_k if and only if $\mathbf{w}$ is a linear combination of $\mathbf{b}_1, ..., \mathbf{b}_k$.

With Theorem 2 in mind, we say that a weighting vector is k-*Borda* if it is a linear combination of $\mathbf{b}_1, ..., \mathbf{b}_k$. Thus, if $j \leq k$, then every j-Borda weighting vector is k-Borda, giving us a natural hierarchy of weighting vectors.

Connection to Pascal's Triangle

When it comes to doing computations, it can often be convenient to replace a given basis of vectors with an orthogonal basis. In our case, it makes sense to orthogonalize the basis $b_1, ..., b_n$ beginning with b_1, then b_2, then b_3, and so on.

If we call the resulting vectors $c_1, ..., c_n$, here is what we get for a few small values of n. For $n = 2$, we get $c_1 = (1, 1)$ and $c_2 = (-1, 1)$. For $n = 3$, we get $c_1 = (1, 1, 1), c_2 = (2, 0, -2)$, and $c_3 = (1, -2, 1)$. For $n = 4$, we get $c_1 = (1, 1, 1, 1), c_2 = (3, 1, -1, -3), c_3 = (3, -3, -3, 3)$ and $c_4 = (-1, 3, -3, 1)$.

With respect to these vectors, there are two ideas we want to highlight right away. First, you would almost certainly not want to use many of these vectors by themselves as the weighting vector for a positional voting procedure. For example, when $n = 4, c_3 = (3, -3, -3, 3)$ assigns more points for coming in last than in second! It is therefore important to keep in mind that these vectors $c_1, ..., c_n$ are useful precisely because they form a useful orthogonal basis.

Second, consider the last vector in each list. These vectors are $(-1, 1), (1, -2, 1)$, and $(-1, 3, -3, 1)$. This seems to suggest that a weighting vector for n candidates is $(n - 1)$-Borda if and only if it is orthogonal to the nth row of Pascal's triangle with alternating signs!

Indeed, this is the case, and this connection between our generalization of the Borda count and Pascal's triangle became crystal clear to us once we realized that our weighting vectors

$b_1, ..., b_n$ are related to what are called Pascal matrices.

For n candidates, the particular Pascal matrix in which we are interested is the lower triangular $n \times n$ matrix whose entries consist of the entries in Pascal's triangle up to level n. For example, if $n = 5$, then we are interested in the following Pascal matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}.$$

One of the most interesting things about a Pascal matrix is that its inverse looks just like itself but with alternating signs! For example, the inverse of the matrix above is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ -1 & 3 & -3 & 1 & 0 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}.$$

How do Pascal matrices help us understand the appearance of those weighting vectors in our study? First of all, it is clear
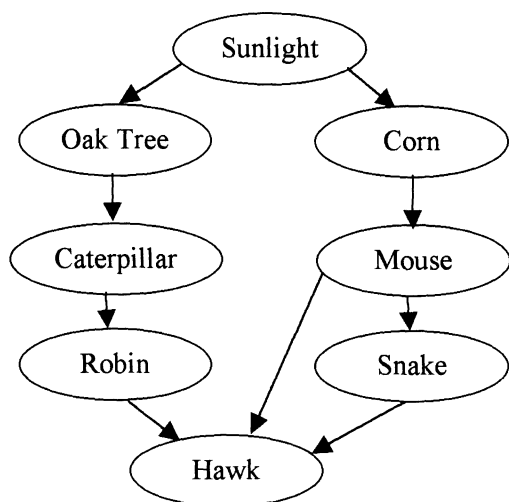
	Su	O	Co	C	M	R	S	H
Su	0	1	1	0	0	0	0	0
O	0	0	0	1	0	0	0	0
Co	0	0	0	0	1	0	0	0
C	0	0	0	0	0	1	0	0
M	0	0	0	0	0	0	1	1
R	0	0	0	0	0	0	0	1
S	0	0	0	0	0	0	0	1
H	0	0	0	0	0	0	0	0

Figure 2. A simple ecosystem and its adjacency matrix. Prey/resources are located at the tails of the arrows and predators/consumers are located at the heads of arrows. Each organism is represented in the matrix by its first letter, except for Sunlight (Su) and Corn (Co). Ecosystem taken from fourth grade TIMSS science test.

goes a significant change. Without this ability to explicitly affect each other, information could not pass through the connected network of individuals, as it would exist only within the computer-generated construct and not within the individual minds connected to the system.

On a related note, the movies take this to the next logical step, illustrated in the way the newly released Agent Smith is able to write himself over other programs in the system. Going further, the Wachowski brothers have Smith copy himself into the brain of an individual who has been freed from the system, thereby releasing himself into the real world. Although this may seem impossible, it is merely an extension of the mechanisms already at work in the computer system. For example, the rebels can easily download information from cartridges into a person's mind in order to teach them kung fu or how to fly a helicopter. This surely involves changes in the real brain

of the individual, for they must remember, at the very least, that they have this knowledge in the computer world in order to make use of it the next time they "plug in." Thus, programming in the computer world can have effects in the real world. Without these reciprocal effects, information—about people and places and abilities—would never pass through the system of interconnected people, and the rebellion would be doomed to failure.

In the end, we are left with not only popular reasons for the movie title, but also mathematical reasons. Without matrices, the computer graphics used to project the images into the minds of the embedded people would not be possible. Matrices also form a vital tool for analyzing the possible paths of future events through the use of Markov chains. And matrices are common tools for representing the connections in a network or agents that share information. ■

that the vectors $(-1, 1), (1, -2, 1), (-1, 3, -3, 1), (1, -4, 6, -4, 1)$, and so on, make up the bottom rows of the inverses of the Pascal matrices. Secondly, notice that the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ appear upside down as the columns of the Pascal matrix.

A moment's thought should convince you that this means that each of $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$ is orthogonal to the (reversal, read right-to-left, of the) last row of the inverse of the Pascal matrix. By construction, however, this must also hold for the vectors $\mathbf{c}_1, \ldots, \mathbf{c}_{n-1}$. So this explains why we were seeing the rows of Pascal's triangle with alternating signs!

The next step in the story would be to try to explain why the other vectors in our orthogonal bases of weighting vectors look the way they do. Fortunately, we have been able to do this by focusing on weighting vectors that are different from $\mathbf{b}_1, \ldots, \mathbf{b}_n$

but that also correspond to simple counting questions. We will save the details for another paper, but we invite you to give it some thought. You may be just as surprised as we were by what you discover! ■

Further Reading

G. Call and D. Velleman, Pascal's matrices, *The American Mathematical Monthly*, Vol. 100, No. 4. (Apr., 1993), 372–376.

Z. Daugherty, A. Eustis, G. Minton, and M. Orrison, Voting, the symmetric group, and representation theory, to appear in *The American Mathematical Monthly*.

D. Saari, *Chaotic elections!*, American Mathematical Society, Providence, RI, 2001.