Krystle Stehno

Susan Gates

Teaching Module on Voting Methods

The purpose of this project is to provide teachers with a module for teaching some basic mathematics about voting. Students do not need to have any prior experience with the subject, but it is recommended for Algebra 1 or in a higher level course. It is also possible to use this module as part of an interdisciplinary unit on government. We estimate that this unit will take a minimum of two days: the first day would cover the background information while the second would cover the voting methods.

We will be using discrete mathematics in this module. The topics include combinations, coloring and some graph theory. We will provide course appropriate material for the students as well as more detailed information for the teacher. We hope that this unit will allow students to see the relationships between math and decision making and that students will take from this unit a broad understanding of the interconnectedness between mathematics and other subject areas/ every day events/ parts of life.

This module is divided into several parts as follows:

1. Background Information
	1. Definitions
	2. Mathematics
		1. Algorithms
		2. C(n,r)
		3. Representing information using a graph and model
2. Types of Voting
	1. Types
	2. Mathematics
		1. Counting the possible outcomes
		2. Modeling the information and determining outcome
	3. Exploring winners in different systems
3. Further Information for teachers and students
4. Homework
5. Resources
6. Bibliography
7. Background Information

In Voting Theory, the goal is to make the largest number of people happy while allowing everyone to vote honestly.

* 1. Definitions:
1. Combination = C(n, r) = the number of r-combinations of an m-set without replacement or repetition
2. Cr(m, r) = C(m + r – 1, r) = the number of r-combinations of an m-set if we sample with replacement or repetition
3. Permutation = P(n, r) = number of ways to pick out r elements from an n-set and arrange them in order
4. P(n, r) = \_n!\_

 (n-r)!

1. C(n, r) = P(n, r)

 P(r, r)

* 1. Mathematics:
		1. Choose $\left(\genfrac{}{}{0pt}{}{n}{r}\right)$ : (Number of ways when order doesn’t matter) Students should understand that this symbol is the same as C(n,r) and what it means, not only the definition (the number of ways to choose r objects given n objects). Students should understand circumstances in which to use this operator (i.e. when determining the number of ways to distribute r gifts among n people, etc).

They should understand how to solve using this operator:

C(n,r) = $\left(\genfrac{}{}{0pt}{}{n}{r}\right)= $ $\frac{n!}{r!\left(n-r\right)!}$

And common solutions including:

$\left(\genfrac{}{}{0pt}{}{n}{n}\right)$ = $\frac{n!}{n!\left(n-n\right)!}$ = $\frac{n!}{n!\left(0\right)!}$ = $\frac{n!}{n!}$ = 1

For r > n : $\left(\genfrac{}{}{0pt}{}{n}{r}\right)$ = 0

* + 1. Algorithms: Students should understand basic algorithms (i.e. be able to give very detailed instructions). To assess this, have students individually or as a class write directions for a simple task, such as making a peanut butter and jelly sandwich or something simpler. Note: the students directions should demonstrate that they are thinking about every step in the process (i.e. open the jar of peanut butter by twisting the lid in a clockwise motion. After removing the lid, put the lid on the table, etc.)
		2. Representing Data for Groups with a Model: Students should be able to model situations by drawing a picture to represent data, and be able to represent the data algebraically if possible.

Given the following situation, students should be able to draw a representation of the situation.

There are ten tables at a restaurant, each with four people. (The expected representation would be a picture of ten tables, each with four people such as follows)

*Restaurant Model*

 P P P P P

P P P P P P P P P P

 P P P P P

 P P P P P

P P P P P P P P P P

 P P P P P

Students should also be able to model situations using charts and understand charts, such as the following.

 *Preferred Ice Cream, ranked in order of preference*

|  |  |  |  |
| --- | --- | --- | --- |
| Person | Chocolate | Vanilla | Strawberry |
| A | 1 | 3 | 2 |
| B | 2 | 1 | 3 |
| C | 3 | 2 | 1 |
| D | 2 | 3 | 1 |
| E | 1 | 3 | 2 |
| F | 1 | 3 | 2 |
| G | 3 | 1 | 2 |
| H | 1 | 2 | 3 |
| I | 3 | 2 | 1 |
| J | 1 | 3 | 2 |

1. Types of Voting

There are single-winner and multiple-winner types of voting, but in our case we are just going to discuss single-winner cases

A voting method contains rules for valid voting and how votes are aggregated to yield a final result. There are many different voting methods including:

* 1. Definitions:
* Plurality: each voter votes for one candidate of his or her choice, majority votes win (single winner)
	+ - 1. Exhaustive ballot: plurality voting in which if no candidate receives an absolute majority of votes then the candidate with the fewest votes is eliminated and another round of voting occurs
			2. Two-round system: plurality voting in which if no candidate receives an absolute majority of votes then all of the candidates except the two with the most votes are eliminated and a second round of voting occurs
* Approval voting: each voter can vote for as many candidates as they wish and the winner is the candidate receiving the most votes. Not preferential voting
* Weighted Voting: each voter is given a certain number of points to divide among all of the candidates
* Borda Count (weighted voting method): allows a voter to rank candidates from best to worst, candidates earn points depending on how they were ranked by each voter. Usually gives n points to top-ranked candidate, (n-1) votes to next… lowest ranked gets 1 point. Candidate with most points wins
* Condorcet Method: uses pair-wise comparisons. Each candidate is compared to every other candidate, and the candidate who wins the most comparisons wins the election
	1. Mathematics
		1. Counting the votes
			1. For **plurality** voting we are using replacement and we can therefore count the number of different patterns of vote totals using C(m + r – 1, r)
			2. With **approval** voting and preferential voting, one winner is again selected, so we have C(m + r -1, r)
			3. For **weighted** voting, there will still only be one winner, so for the number of outcomes, we have the same as above. If we are looking at the possible point break downs, the problem would be much more difficult. We would have n points to divide among m candidates and thereby outcomes ranging from one person getting nq, if q is the number of points and everyone else getting none to a tie between all candidates.
			4. The total number of points in a **Borda Count** election (with 1 pt. for the last place candidate, etc.) is given by
			(sum of the point value for each place)(number of voters).
			5. In the **Condorcet Method**, we first find the number of possible pairs. If there are n candidates, there would be $\frac{n-2}{2}$ pairs. From here, the person with the most pair-wins is the winner. But again, if there are n candidates, we have the same number of possibilities as above.
		2. Modeling
			1. Plurality Voting: We can model our plurality voting using a bipartite graph. Our vertices would be candidates on one side and voters on the other. Our edges would connect the person voting to their number one choice. In this case the degree of each voter vertex would be one. The winner would be determined by the candidate with the largest degree.
			2. Approval Voting: We could model this with a bipartite graph like we did with plurality voting, the only difference being that each person could have multiple edges standing for their multiple votes.
			3. Borda Count (weighted voting): We could model this with the bipartite graph described for approval voting except each edge has a different weight. The vertices for the voters would each have a degree equal to the number of candidates. In order to find the winner we would add up the weights of all of the edges connected to each candidate separately, and the candidate with the highest weight would win. This model would be confusing and time consuming for large data sets, and it would be easier to find the winner just by adding up ranks from a data table.
			4. Condorcet method: see the higher level concepts for teachers and advanced students section.
	2. In order to explore the different outcomes for different voting methods, we suggest using the Appendix 1 model.

3) Higher level concepts for teachers and advanced students:

* 1. Arrow’s Theorem

Formally stated, Arrow’s theorem is as follows:

Let A be a set of outcomes, N a number of voters or decision criteria. The set of all full linear orderings of A is then denoted by by L(A) . Note: This set is equivalent to the set S | A | of permutations on the elements of A).

A social welfare function is a function, F: L(A)N→L(A), which aggregates voters' preferences into a single preference order on A. The N-tuple (R1, …, RN) of voter's preferences is called a preference profile. In its strongest and most simple form, Arrow's impossibility theorem states that whenever the set A of possible alternatives has more than 2 elements, then the following three conditions, called fairness criteria become incompatible:

* Unanimity (Pareto efficiency)

If alternative a is ranked above b for all orderings R1, …, RN, then a is ranked higher than b by F(R1, …, RN). (Note that unanimity implies non-imposition).

* Non-dictatorship

There is no individual i whose preferences always prevail. That is, there is no

 i є {1, …,N} such that for every (R1, …, RN ) є L(A)N , F(R1, …, RN) = Ri..

* [Independence of Irrelevant Alternatives](http://en.wikipedia.org/wiki/Independence_of_irrelevant_alternatives)

For two preference profiles (R1, …, RN) and (S1, …, SN) such that for all individuals i, alternatives a and b have the same order in Ri as in Si, alternatives a and b have the same order in F(R1, …, RN) as in F(S1, …, SN).

"There is no consistent method by which a democratic society can make a choice (when voting) that is always fair when that choice must be made from among 3 or more alternatives." .

The theorem is commonly restated as:"No voting method is fair", "Every ranked voting method is flawed", or "The only voting method that isn't flawed is a dictatorship". But these are oversimplified, and thus do not hold universally.

The theorem actually says that a voting mechanism can’t follow all the fairness criteria for all possible preference orders (i.e. any social choice system respecting unrestricted domain, unanimity, and independence of irrelevant alternatives is a dictatorship).

The theorem has been proven in a number of ways, several of which use discrete mathematics. In this module, we have provided a simply graph theory proof of the theorem. There are additional proofs, some of which use discrete mathematics, which can be found in the bibliography.

Simple Graph Theory Proof of Arrow’s Theorem: from a paper by Nambiar, Varma and Saroch, submitted May 1992. *<* www.ece.rutgers.edu/~knambiar/science/ArrowProof.pdf*>*

This proof uses two directed graphs D=(V,A): a preference and a nonpreference digraph. Nonpreference digraph is complete and transitive. Complete in this case is used to mean that there exists an arc between every pair of vertices. Transitive means that if the arcs (a,b) and (b,c) exist then the arc (a,c) exists. Preference graph is the complement of the nonpreference graph. The preference graph is assymetrical and transitive. Adjacency matrix of preference graph is called the preference matrix.

The notation that will be used in the proof is as follows:

m is the total number of candidates C1, …….., Cm

n is the total number of voters V1, …….., Vn

Vk  = [vi, jk ] is the preference matrix of order m by m which gives the preference of the voter, Vk, where k is an element of {1, …, n}. When vi, jk = 0, the voter does not prefer candidate i over candidate j. When vi,jk = 1, the voter prefers candidate i over candidate j. **0** (boldface) represents a nonpreference set of voters, while **1** (also boldface) represents a preference set of voters.

vi, jk = \* means that the voter has an unspecified preference. The star also represents the unspecified preference set of voters.

S = [si j ] is the preference matrix of m by m order which gives the preference of society as a whole, rather than individual voters. The voting function is F(V1, …….., Vn) = S. The dictator function is also a projection function and is as follows: Dnk (x1, ….xn)= xk .

Two axioms are used in the proof. The ideas were previously mentioned as part of the fairness criteria which must be observed. They are the axiom of independence which states that sij  is a function of the vijk ‘s only. It is as follows: Sij= fij(vij1,……..vijn) for i≠j and sij=0 .

The second axiom is the axiom of unanimity that states that if all the voters vote one way then the voting system also votes the same way (definition of unanimous). It is as follows: fij(0,0,…..0)=fij(1,1,…..1)=1.

We want to prove: fij(x1, …..xn)= Dnd(x1, ….xn)= xd which means that S=Vd. Note: x is either the candidate or the voter, as long as it is the same in both f and D.

Proof:

Define h= minij {$ \sum\_{k=1}^{n}x\_{k } | f\_{ij}(x\_{1}, …..x\_{n})=1 $}

Note that the m(m-1)2n values of fij need to be examined before the value of

H can be obtained. We want to show that h = 1.

fij(1; 0; 0) = 0 and fjk(1; 0; 0) = 0

→fik (\*,\*,0) = 0 since nonpreference graphs are transitive

→ fik(1; 1; 0) = 0

Taking the contrapositive of the above argument

fik (1; 1; 0) = 1

→ fij(1; 0; 0) = 1 or fjk(0; 1; 0) = 1

It immediately follows that h = 1. Note that h cannot be zero because of the unanimity

axiom.

Without loss of generality we may assume fab(1; 0) = 1, the position at which 1 occurs

in fab is of no concern to us. Here Ca and Cb are two specific candidates. Now,

fia(1; 1) = 1 and fab(1; 0) = 1

→fib(1; \*) = 1 since preference graphs are transitive, and

fib(1; \*) = 1 and fbj(1; 1) = 1

→ fij(1;\*) = 1 since preference graphs are transitive

→ fji(0; \*) = 0 since preference graphs are asymmetric

→ fij(x1; \*) = x1

Dictator Theorem immediately follows.

1. An advanced overview of Condorcet Voting:
* Condorcet Voting (Single-Winner): All candidates are ranked and compared in pair-wise elections, whoever has the most wins is elected.
* The actual voting process:
	+ In a Condorcet election the voter ranks the list of candidates in order of preference (for example, the voter gives a 1 to their first preference, a 2 to their second preference…)
	+ When a voter does not give a full list of preferences they are assumed to prefer the candidates they have ranked over all other candidates.
	+ Finding the Winner:
		- The count is conducted by putting every candidate against every other candidate in a series of imaginary one-on-one contests. The winner of each pairing is the candidate preferred by a majority of voters.
* Modeling the Condorcet Method with a Digraph:
	+ Given a voting profile for an election with *n* candidates, its corresponding Condorcet digraph *G* = (*V, E*) has one vertex for each of the *n* candidates. For each candidate pair (*x, y*), there exists an edge from *x* to *y* (denoted by *x → y*) if *x* would receive at least as many votes as *y* in a head-to-head contest. In other words, *x → y* if *x* is ranked above *y* by at least as many voters as ranked *y* above *x*. For the candidates that tie there is an edge pointing in each direction (denoted *x ↔ y*).
	+ The Condorcet digraph of any profile contains at least one edge between every pair of candidates. We call digraphs with at least one edge between any two nodes *semi-complete*.
	+ Any candidate that beats or ties with all others is called a Condorcet winner. In the Condorcet digraph, this corresponds to having an out-degree of *n –* 1
* The Condorcet Voting Algorithm: The Condorcet voting algorithm is a majoritarian method which specifies that the winner of the election is the candidate(s) that beats or ties with every other candidate in a pair-wise comparison
	+ - **Algorithm 1** Simple Majority Runoff.
		- 1: *count* = 0
		- 2: **for** each of the *k* search systems *Si***do**
		- 3: If *Si* ranks *d*1 above *d*2, *count*++
		- 4: If *Si* ranks *d*2 above *d*1, *count−−*
		- 5: If *count >* 0, rank *d*1 better than *d*2
		- 6: Else rank *d*2 better than *d*1
* Condorcet MetaSearch: We can generalize the notion of our winner being the winner of all pair-wise contests to generate a ranked list of candidates by modeling the election with a directed graph of candidates which we call the Condorcet digraph. A Hamiltonian traversal of this digraph will produce the election rankings. Algorithm 2 shows an instantiation of this idea to metasearch.
	+ To generate the digraph takes *O(n2k)* time where *n* is the number of documents in the metasearch pool and *k* is the number of search engines, which makes it too slow for real applications.
		- **Algorithm 2:** Theoretic Condorcet Metasearch.
		- 1: **for** all pairs (*d*1*,d*2) of documents **do**
		- 2:Use Algorithm 1 to compute the Condorcet Digraph
		- 3: Compute Hamiltonian path of Condorcet digraph
* Condorcet’s Paradox: Condorcet’s paradox is a situation in which collective preferences can by cyclic. This means that majority wishes can be in conflict with each other. When this occurs it is because the conflicting majorities are each made up of different groups of individuals.
	+ Example: Voter 1: A B C

 Voter 2: B C A

 Voter 3: C A B

If C is chosen as the winner, it can be argued that B should win instead since two voters prefer B to C and only one voter prefers C to B.

By the same argument, A is preferred to B and C is preferred to A. Thus, there is no clear winner.

* Strongly Connected Components of the Condorcet Digraph:
	+ We need to talk about strongly connected components of a Condorcet digraph so that we can avoid the Condorcet voting paradox since the SCC digraphs are acyclic by design.
	+ The strongly connected components (SCC) of a digraph partition the vertices of the digraph.
	+ Two vertices are in the same equivalence class if there exists a cycle in the digraph containing them.
	+ Each SCC of the Condorcet digraph contains a set of equivalent nodes. In the strongly connected component digraph (SCCD) of a digraph, each node represents one SCC and for X, Y in the SCCD, *X → Y if there exists an x ∈ X* and *y ∈ Y* such that *x → y.*
	+ In the case of semi-complete digraphs, if *X → Y* , then *xi → yj* for all *xi∈ X* and *yj∈ Y*
	+ Since the original digraph was semi-complete, so is the SCCD. Since the SCCD is acyclic, this implies that there is one node X with in-degree zero. If X is removed from the SCCD, we still have a semi-complete, acyclic digraph. Therefore, this digraph has node X’ with in-degree zero. This process can be repeated until all nodes have been exhausted. This creates a unique ordering.
	+ Each SCC represents a pocket of cycles within which we say each candidate is tied. The SCC’s partition the digraph into sets of tied candidates.
	+ Example:



In this case we would be left with a tie!

The out-degree of Strawberry is 2 and the out-degree of Chocolate is 2, resulting in no clear winner.

* Condorcet Paths:
	+ We define a *Condorcet-consistent Hamiltonian path* (or *Condorcet path*) to be any Hamiltonian path through the Condorcet digraph. Our goal is to efficiently find such a path. Condorcet paths have two properties relevant to metasearch:
	+ Theorem 1. *Every semi-complete graph contains a Hamiltonian path.*
		- Proof by induction.
			* The base case is a graph with one node and is trivial.
			* For the inductive step, suppose every semi-complete graphwith *n−*1 nodes contains a Hamiltonian path. For a problem of size *n*, let *H* be the Hamiltonian path for a sub-problem containing only an arbitrary *n−*1 of the nodes. Now the *nth* node *x* is introduced, along with its *n −* 1 edges to or from the other nodes. There are three cases: (1) If *x* points to the first node in *H*, then *x* followed by *H* is a Hamiltonian path. (2) If not, then the first node in *H* points to *x*. Now consider each node in *H* in turn. A new Hamiltonian path can be created by inserting *x* into *H* just before the first node that *x* points to, if one exists. (3) If *x* doesn’t point to any of the nodes in *H*, then the last node in *H* points to *x*, so a new Hamiltonian path can be created by appending *x* to *H*.
		- A Picture Proof:



* Theorem 2*. If candidate x is in an SCC ranked above the SCC containing y, then x is ranked above y in every Condorcet path.*
	+ Proof. This is a simple consequence of the fact that there is only one way to sort the SCCD (strongly connected component digraph); thus there does not exist a path from *y* to *x*. So any Hamiltonian path puts *x* before *y*. Thus we say Condorcet paths *properly order* the SCC’s. In other words, a randomly chosen Condorcet path orders the candidates that don’t tie correctly, and breaks ties arbitrarily. We now know that we can find a unique ordering by generating the entire Condorcet digraph, computing it’s SCC’s, sorting them and ordering candidates within each SCC arbitrarily. As stated earlier, this algorithm is *O(n2k)* for *n* candidates and *k* voters, which is impractical for large sets.
* Theorems 1 and 2 show that we can find a reasonable ordering by computing a Condorcet path.
* Condorcet-fuse Algorithm:
	+ The Condorcet-fuse algorithm does exactly that for us, giving us an algorithm that finds a Condorcet path in *O(nk lgn)* time without actually creating the Condorcet digraph.
	+ The key is to use Algorithm 1 as the comparison function in the InsertionSort Algorithm. This could also be used with MergeSort or QuickSort as well.
		- **Algorithm 3** Condorcet-fuse.
		- 1: Create a list *L* of all the documents
		- 2: Sort(*L*) using Algorithm 1 as the comparison function
		- 3: Output the sorted list of document

4) Homework

For the homework, see Appendix 2.

5) Additional Resources:

* National Council of Teachers of Mathematics Website: <http://www.nctm.org>
	+ Sets the standards for math curriculum in the United States, has many teacher resources like lessons, standards, problems, activities, tips, and articles for mathematics
* PBS Teachers Website: <http://www.pbs.org/teachers>
	+ A very good website with examples and problems sorted by grade range and topic
* Teachnology Website: <http://www.teach-nology.com/teachers>
	+ Gives lesson plans, worksheets, tips and tools for all subjects
* <http://www.colorado.edu/education/DMP/activities/index.html>
	+ Gives a lot of activities for voting theory

The following books have chapters that discuss various voting methods:

* Tannenbaum, P., & Arnold, R. (1997). *Excursions in Modern Mathematics*. Upper Saddle River, NJ: Prentice Hall.
* Consortium for Math and Its Applications (COMAP). (2006). *For All Practical Purposes: Mathematical Literacy in Today’s World, 7th ed*. New York: W. H. Freeman.
* The following websites have addition proof of Arrow’s Theorem:
	+ http://www.indiana.edu/~econed/pdffiles/summer02/phansen.pdf
	+ http://ideas.repec.org/p/cwl/cwldpp/1123r3.html
	+ http://en.wikipedia.org/wiki/Arrow%27s\_impossibility\_theorem
1. Bibliography:

Alon, Noga. Voting paradoxes and digraphs realizations. Advances in Applied Mathematics, Volume 29, Issue 1, July 2002, Pages 126-135.

Aslam, J, & Montague, M (2002). Condorcet Fusion for Improved Retrieval. *ACM*, *1-58113-492-4/02/0011*, Retrieved April 26, 2009, from <http://www.ccs.neu.edu/home/jaa/ISU535.05X1/resources/condorcet.pdf.>

Beigman, Eyal. “Extension of Arrow’s Theorem to Symmetric Sets of Tournaments.” Discrete Mathematics. Vol 301 pg 2074-2081.

Brown, Richard and Mary P. Dolciani, Robert H. Sorgenfrey. Algebra 1: Explorations and Applications. McDougal Littell:1997

Crowley, Mark (2006, January 30). Election Theory. Retrieved April 26, 2009, Web site: <http://www.cs.ubc.ca/~crowley/academia/papers/gtdt-election-jan302006.pdf>

“Discrete Mathematics Links.” <http://www.math.uncc.edu/~hbreiter/problems/dmlinks.html>

“The Mathematics of Voting.”< http://www.ctl.ua.edu/math103/Voting/mathemat.htm>

VOLUME 81, NUMBER 8 PHYSICAL REVIEW LETTERS 24 AUGUST 1998

Saari, Donald G. Which is better: the Condorcet or Borda winner? Received: 26 April 2004 / Accepted: 4 June 2004 / Published online: 7 January 2006 © Springer-Verlag 2006

Voting Procedures by Steven J. Brams and Peter C. Fishburn, October 1998 New York University

Voting power when using preference ballots Deanna B. Haunsperger 1, Duncan J. Melville Carleton College, Northfield, MN 55057, USA Received: 2 March 1995/Accepted: 7 September 1995

Is the Act of Voting Rational? By Yoram Barzel and Eugene Silberberg

Discrete Mathematics in Voting and Group Choice by Peter C. Fishburn, ©1984 Society for Industrial and Applied Mathematics.

 (2009, April 20). Condorcet Method. Retrieved April 26, 2009, from Wikipedia Web site: <http://en.wikipedia.org/wiki/Condorcet\_method#Voting >

Namblar, K.K., Prarnod K. Varma and Vandana Saroch. “A Graph Theoretic Proof of Arrow’s Dictator Theorem.” May 1992.

Powers, R.C. “Arrow’s Theorem for Cloesd Weak Hierarchies.” Discrete Applied Mathematics. Vol 66 pg 271-278.