Decomposition Behavior in Aggregated Data Sets

Sarah Berube Karl-Dieter Crisman

Gordon College

Oct. 24, 2009

Outline

Outline

Paradox in non-parametric statistics

Aggregation can be a source of paradox in statistics. Here is a simple (Yule-)Simpson-like example:

Paradox in non-parametric statistics

Aggregation can be a source of paradox in statistics. Here is a simple (Yule-)Simpson-like example:

Example

Imagine the following stores convinced x out of y (x/y) customers to buy something on the following days:

	Day 1	Day 2	Total
Store 1	2/3	7/20	9/23
Store 2	9/20	1/3	10/23

Even though Store 1 has a better success rate on both days, the aggregate data suggests that Store 2 was actually better at luring customers to buy.

Paradox in non-parametric statistics

Aggregation can be a source of paradox in statistics. Here is a simple (Yule-)Simpson-like example:

Example

Imagine the following stores convinced x out of y (x/y) customers to buy something on the following days:

	Day 1	Day 2	Total
Store 1	2/3	7/20	9/23
Store 2	9/20	1/3	10/23

Even though Store 1 has a better success rate on both days, the aggregate data suggests that Store 2 was actually better at luring customers to buy.

The point is, aggregation of data can yield unexpected results, and that is particularly true when looking solely at ranking of data.

Paradox in non-parametric statistics

In the last two decades, tools from the mathematics of voting have been used to begin to unravel some of these paradoxes. Haunsperger specifically addresses aggregation paradoxes in her 2003 *Social Choice and Welfare* paper, from whence we draw our first example.

Paradox in non-parametric statistics

In the last two decades, tools from the mathematics of voting have been used to begin to unravel some of these paradoxes. Haunsperger specifically addresses aggregation paradoxes in her 2003 *Social Choice and Welfare* paper, from whence we draw our first example.

First, we recall the Kruskal-Wallis test. As a uniformity test for data samples with three populations, it may be viewed as follows:

Paradox in non-parametric statistics

In the last two decades, tools from the mathematics of voting have been used to begin to unravel some of these paradoxes. Haunsperger specifically addresses aggregation paradoxes in her 2003 *Social Choice and Welfare* paper, from whence we draw our first example.
First, we recall the Kruskal-Wallis test. As a uniformity test for data samples with three populations, it may be viewed as follows:

- Take sampling data for each population and organize it in a table.

Paradox in non-parametric statistics

In the last two decades, tools from the mathematics of voting have been used to begin to unravel some of these paradoxes. Haunsperger specifically addresses aggregation paradoxes in her 2003 *Social Choice and Welfare* paper, from whence we draw our first example.
First, we recall the Kruskal-Wallis test. As a uniformity test for data samples with three populations, it may be viewed as follows:

- Take sampling data for each population and organize it in a table.
- Replace the data by the rank order of the data, smallest to largest.

Paradox in non-parametric statistics

In the last two decades, tools from the mathematics of voting have been used to begin to unravel some of these paradoxes. Haunsperger specifically addresses aggregation paradoxes in her 2003 *Social Choice and Welfare* paper, from whence we draw our first example.

First, we recall the Kruskal-Wallis test. As a uniformity test for data samples with three populations, it may be viewed as follows:

- Take sampling data for each population and organize it in a table.
- Replace the data by the rank order of the data, smallest to largest.
- Sum the columns of the ranks and determine whether they are too dissimilar to be from identical populations.

Paradox in non-parametric statistics

In the last two decades, tools from the mathematics of voting have been used to begin to unravel some of these paradoxes. Haunsperger specifically addresses aggregation paradoxes in her 2003 *Social Choice and Welfare* paper, from whence we draw our first example.

First, we recall the Kruskal-Wallis test. As a uniformity test for data samples with three populations, it may be viewed as follows:

- Take sampling data for each population and organize it in a table.
- Replace the data by the rank order of the data, smallest to largest.
- Sum the columns of the ranks and determine whether they are too dissimilar to be from identical populations.
- Alternately, one may view this as giving a 'ranking' of the populations.

Haunsperger's Example

- Consider the two sets of data

A	B	C
5.89	5.81	5.80
5.98	5.90	5.99

and

A	B	C
5.69	5.63	5.62
5.74	5.71	6.00

Haunsperger's Example

- Consider the two sets of data

A	B	C
5.89	5.81	5.80
5.98	5.90	5.99

and

A	B	C
5.69	5.63	5.62
5.74	5.71	6.00

- Both will give rise to the **same** *matrix of ranks*,

A	B	C
3	2	1
5	4	6

.

  The column sums are 8, 6, and 7.

Haunsperger's Example

- Consider the two sets of data

A	B	C
5.89	5.81	5.80
5.98	5.90	5.99

and

A	B	C
5.69	5.63	5.62
5.74	5.71	6.00

- Both will give rise to the **same** *matrix of ranks*,

A	B	C
3	2	1
5	4	6

.

  The column sums are 8, 6, and 7.

  Combining the two sets gives the following matrix of ranks, which has column sums 26, 22,

A	B	C
8	7	6
10	9	11
3	2	1
5	4	12

- and 30 - so that not only are the differences more pronounced, but C seems now to be the population with the 'biggest' result.

Recent Work

As it turns out, this is not unusual behavior.

- Haunsperger shows that nearly all data sets are to some extent inconsistent under such aggregation for Kruskal-Wallis.

Recent Work

As it turns out, this is not unusual behavior.

- Haunsperger shows that nearly all data sets are to some extent inconsistent under such aggregation for Kruskal-Wallis.
- Bargagliotti (2009) extends this to the whole class of such tests.

Recent Work

As it turns out, this is not unusual behavior.

- Haunsperger shows that nearly all data sets are to some extent inconsistent under such aggregation for Kruskal-Wallis.
- Bargagliotti (2009) extends this to the whole class of such tests.
- On the other hand, Bargagliotti and Greenwell show that the statistical significance of current results is negligible.

Recent Work

As it turns out, this is not unusual behavior.

- Haunsperger shows that nearly all data sets are to some extent inconsistent under such aggregation for Kruskal-Wallis.
- Bargagliotti (2009) extends this to the whole class of such tests.
- On the other hand, Bargagliotti and Greenwell show that the statistical significance of current results is negligible.

And, one can analyze these things using voting theory!

Recent Work

As it turns out, this is not unusual behavior.

- Haunsperger shows that nearly all data sets are to some extent inconsistent under such aggregation for Kruskal-Wallis.
- Bargagliotti (2009) extends this to the whole class of such tests.
- On the other hand, Bargagliotti and Greenwell show that the statistical significance of current results is negligible.

And, one can analyze these things using voting theory!

- Many nonparametric procedures create a test statistic by a method equivalent to first creating a voting profile, to which standard procedures are applied. (This is Haunsperger and Saari's approach.)

Recent Work

As it turns out, this is not unusual behavior.

- Haunsperger shows that nearly all data sets are to some extent inconsistent under such aggregation for Kruskal-Wallis.
- Bargagliotti (2009) extends this to the whole class of such tests.
- On the other hand, Bargagliotti and Greenwell show that the statistical significance of current results is negligible.

And, one can analyze these things using voting theory!

- Many nonparametric procedures create a test statistic by a method equivalent to first creating a voting profile, to which standard procedures are applied. (This is Haunsperger and Saari's approach.)
- Hence, looking at a decomposition of the profile vector with respect to a useful basis could help! Work in this direction is begun in Bargagliotti and Saari (2007); for instance, criteria for avoiding certain paradoxes is given.

Basics under aggregation

- The component of any decomposition which yields the fewest paradoxes is called the Basic component.

Basics under aggregation

- The component of any decomposition which yields the fewest paradoxes is called the Basic component.
- From a theoretical viewpoint, it is useful to look at the most consistent situation first.

Basics under aggregation

- The component of any decomposition which yields the fewest paradoxes is called the Basic component.
- From a theoretical viewpoint, it is useful to look at the most consistent situation first.
- So we raise the following questions regarding the Basic component:

Questions

Basics under aggregation

- The component of any decomposition which yields the fewest paradoxes is called the Basic component.
- From a theoretical viewpoint, it is useful to look at the most consistent situation first.
- So we raise the following questions regarding the Basic component:

Questions

- *How does it behave under aggregation, or at least under replication?*

Basics under aggregation

- The component of any decomposition which yields the fewest paradoxes is called the Basic component.
- From a theoretical viewpoint, it is useful to look at the most consistent situation first.
- So we raise the following questions regarding the Basic component:

Questions

- *How does it behave under aggregation, or at least under replication?*
- *How close can we come to a data set with no other components?*

Basics under aggregation

- The component of any decomposition which yields the fewest paradoxes is called the Basic component.
- From a theoretical viewpoint, it is useful to look at the most consistent situation first.
- So we raise the following questions regarding the Basic component:

Questions

- *How does it behave under aggregation, or at least under replication?*
- *How close can we come to a data set with no other components?*
- *How might one recognize such a data set?*

Basics under aggregation

- The component of any decomposition which yields the fewest paradoxes is called the Basic component.
- From a theoretical viewpoint, it is useful to look at the most consistent situation first.
- So we raise the following questions regarding the Basic component:

Questions

- *How does it behave under aggregation, or at least under replication?*
- *How close can we come to a data set with no other components?*
- *How might one recognize such a data set?*

- We answer many of these questions in this talk.

Outline

Data Definitions

We will need a number of definitions before proceeding.

- We have already encountered a *data set* and the corresponding *matrix of ranks*:

A	B	C
14.5	15.6	16.7
14.3	11.2	13.4

A	B	C
4	5	6
3	1	2

- We can then create a *profile* and *profile vector*.
  - Look at all possible triplets of ranks (one for each item) and, for each of these triplets, return the ranking of the items corresponding to that.
  - In this example, we can see that $(4\,1\,2)$ would correspond to $A \succ C \succ B$, while $(4\,1\,6)$ gives $C \succ A \succ B$, and so on.
  - Our example gives $(0, 2, 2, 2, 0, 2)$, using the usual order $A \succ B \succ C, A \succ C \succ B, \ldots, B \succ A \succ C$.

Components

We use the standard irreducible symmetric decomposition from *Basic Geometry of Voting*, and more recently Orrison et al.:

- The Basic components, $B_A = (1, 1, 0, -1, -1, 0)$, $B_B = (0, -1, -1, 0, 1, 1)$, and $B_C = (-1, 0, 1, 1, 0, -1)$.
- The Reversal components $R_A = (1, 1, -2, 1, 1, -2)$, $R_B = (-2, 1, 1, -2, 1, 1)$, and $R_C = (1, -2, 1, 1, -2, 1)$.

Components

We use the standard irreducible symmetric decomposition from *Basic Geometry of Voting*, and more recently Orrison et al.:

- The Basic components, $B_A = (1, 1, 0, -1, -1, 0)$, $B_B = (0, -1, -1, 0, 1, 1)$, and $B_C = (-1, 0, 1, 1, 0, -1)$.
- The Reversal components $R_A = (1, 1, -2, 1, 1, -2)$, $R_B = (-2, 1, 1, -2, 1, 1)$, and $R_C = (1, -2, 1, 1, -2, 1)$. (Note that they have the same algebraic structure as the Basic profiles, over Σ_3.)

Components

We use the standard irreducible symmetric decomposition from *Basic Geometry of Voting*, and more recently Orrison et al.:

- The Basic components, $B_A = (1, 1, 0, -1, -1, 0)$, $B_B = (0, -1, -1, 0, 1, 1)$, and $B_C = (-1, 0, 1, 1, 0, -1)$.
- The Reversal components $R_A = (1, 1, -2, 1, 1, -2)$, $R_B = (-2, 1, 1, -2, 1, 1)$, and $R_C = (1, -2, 1, 1, -2, 1)$.
- The Condorcet component $C = (1, -1, 1, -1, 1, -1)$.
- The Kernel component $K = (1, 1, 1, 1, 1, 1)$ measures the number of voters.

Components

We use the standard irreducible symmetric decomposition from *Basic Geometry of Voting*, and more recently Orrison et al.:

- The Basic components, $B_A = (1, 1, 0, -1, -1, 0)$, $B_B = (0, -1, -1, 0, 1, 1)$, and $B_C = (-1, 0, 1, 1, 0, -1)$.
- The Reversal components $R_A = (1, 1, -2, 1, 1, -2)$, $R_B = (-2, 1, 1, -2, 1, 1)$, and $R_C = (1, -2, 1, 1, -2, 1)$.
- The Condorcet component $C = (1, -1, 1, -1, 1, -1)$.
- The Kernel component $K = (1, 1, 1, 1, 1, 1)$ measures the number of voters.

In our example, we get

$$\begin{pmatrix} 4 & 5 & 6 \\ 3 & 1 & 2 \end{pmatrix} \Rightarrow (0, 2, 2, 2, 0, 2) \Rightarrow (-1/3, -2/3, -1/3, 0, -2/3, 4/3)$$

which can be written $\frac{1}{3}(-B_A - 2B_B - R_A - 2C + 4K)$.

Aggregation Definitions

Haunsperger provides useful definitions, for a given statistical procedure whose outcome is ranking of the candidates, and for all matrices of ranks:

- The procedure is *consistent under aggregation* if any aggregate of k sets of data, all of which yield a given ordering of the candidates, also yields the same ordering.
- The procedure is *consistent under replication* if any aggregate of k sets of data, all of which have the same matrix of ranks, yields the same ordering as any individual data set.

In the sequel, our concern is with a specific form of replication, which we call *stacking*.

Outline

Defining Stacking

- Stacking is aggregating k data sets, all of which have the same matrix of ranks, and which in addition do not have any overlap between the numerical ranges of their data.

Defining Stacking

- Stacking is aggregating k data sets, all of which have the same matrix of ranks, and which in addition do not have any overlap between the numerical ranges of their data.
- We stack our original example, with $k = 3$: $\left(\begin{array}{ccc} 16 & 17 & 18 \\ 15 & 13 & 14 \\ \hline 10 & 11 & 12 \\ 9 & 7 & 8 \\ \hline 4 & 5 & 6 \\ 3 & 1 & 2 \end{array}\right)$

Defining Stacking

- Stacking is aggregating k data sets, all of which have the same matrix of ranks, and which in addition do not have any overlap between the numerical ranges of their data.
- We stack our original example, with $k = 3$: $$\left(\begin{array}{ccc} 16 & 17 & 18 \\ 15 & 13 & 14 \\ \hline 10 & 11 & 12 \\ 9 & 7 & 8 \\ \hline 4 & 5 & 6 \\ 3 & 1 & 2 \end{array}\right)$$
- Each part of the matrix corresponding to the original matrix of ranks we will call a *stanza*, and we will typically delineate the stanzas.

Defining Stacking

- Stacking is aggregating k data sets, all of which have the same matrix of ranks, and which in addition do not have any overlap between the numerical ranges of their data.
- We stack our original example, with $k = 3$: $\left(\begin{array}{ccc} 16 & 17 & 18 \\ 15 & 13 & 14 \\ \hline 10 & 11 & 12 \\ 9 & 7 & 8 \\ \hline 4 & 5 & 6 \\ 3 & 1 & 2 \end{array}\right)$
- Each part of the matrix corresponding to the original matrix of ranks we will call a *stanza*, and we will typically delineate the stanzas.
- A naive idea of how this might occur is taking samples of the same things, but before and after some big event.

Defining Stacking

- Stacking is aggregating k data sets, all of which have the same matrix of ranks, and which in addition do not have any overlap between the numerical ranges of their data.
- We stack our original example, with $k = 3$: $\begin{pmatrix} 16 & 17 & 18 \\ 15 & 13 & 14 \\ \hline 10 & 11 & 12 \\ 9 & 7 & 8 \\ \hline 4 & 5 & 6 \\ 3 & 1 & 2 \end{pmatrix}$
- Each part of the matrix corresponding to the original matrix of ranks we will call a *stanza*, and we will typically delineate the stanzas.
- A naive idea of how this might occur is taking samples of the same things, but before and after some big event.
  - Prices before and after a huge tax increase

Defining Stacking

- Stacking is aggregating k data sets, all of which have the same matrix of ranks, and which in addition do not have any overlap between the numerical ranges of their data.
- We stack our original example, with $k = 3$: $\left(\begin{array}{ccc} 16 & 17 & 18 \\ 15 & 13 & 14 \\ \hline 10 & 11 & 12 \\ 9 & 7 & 8 \\ \hline 4 & 5 & 6 \\ 3 & 1 & 2 \end{array}\right)$
- Each part of the matrix corresponding to the original matrix of ranks we will call a *stanza*, and we will typically delineate the stanzas.
- A naive idea of how this might occur is taking samples of the same things, but before and after some big event.
  - Prices before and after a huge tax increase
  - Animal populations before and after a conservation effort.

Decomposing Profiles from Stacks of Ranks

We are now ready to completely answer the first question about basics, with respect to stacking.

Decomposing Profiles from Stacks of Ranks

We are now ready to completely answer the first question about basics, with respect to stacking.

Theorem

If we stack an $n \times 3$ matrix of ranks k times, each Basic component is multiplied by k^2, each Reversal component is multiplied by k, the Condorcet component is multiplied by k^2, and the Kernel component is multiplied by k^3.

Decomposing Profiles from Stacks of Ranks

We are now ready to completely answer the first question about basics, with respect to stacking.

Theorem

If we stack an $n \times 3$ matrix of ranks k times, each Basic component is multiplied by k^2, each Reversal component is multiplied by k, the Condorcet component is multiplied by k^2, and the Kernel component is multiplied by k^3.

The implication is that as long as you start with a Condorcet component smaller than the Basic components, stacking is a good way to find data sets with very large Basic components (and hence great regularity in outcome with respect to a variety of procedures).

Decomposing Profiles from Stacks of Ranks

At least for this sort of aggregation, we can avoid some paradox. We have several immediate corollaries:

Decomposing Profiles from Stacks of Ranks

At least for this sort of aggregation, we can avoid some paradox. We have several immediate corollaries:

Corollary

The Kruskal-Wallis test is consistent under stacking, as are any procedures (such as Mann-Whitney) which only rely on pairwise data.

Decomposing Profiles from Stacks of Ranks

At least for this sort of aggregation, we can avoid some paradox. We have several immediate corollaries:

Corollary

The Kruskal-Wallis test is consistent under stacking, as are any procedures (such as Mann-Whitney) which only rely on pairwise data.

(This is because the K-W test, since it comes from the Borda Count, only obeys the Basic component, and in general the Condorcet and Basic components will always be in the same proportion.)

Decomposing Profiles from Stacks of Ranks

At least for this sort of aggregation, we can avoid some paradox. We have several immediate corollaries:

Corollary

The Kruskal-Wallis test is consistent under stacking, as are any procedures (such as Mann-Whitney) which only rely on pairwise data.

Corollary

All tests derived from points-based voting procedures (such as the V test) are consistent under stacking of data sets with no Reversal component.

Decomposing Profiles from Stacks of Ranks

At least for this sort of aggregation, we can avoid some paradox. We have several immediate corollaries:

Corollary

The Kruskal-Wallis test is consistent under stacking, as are any procedures (such as Mann-Whitney) which only rely on pairwise data.

Corollary

All tests derived from points-based voting procedures (such as the V test) are consistent under stacking of data sets with no Reversal component.

(These procedures only differ when it comes to the Reversal component, and otherwise the same argument about Condorcet and Borda applies.)

Decomposing Profiles from Stacks of Ranks

At least for this sort of aggregation, we can avoid some paradox. We have several immediate corollaries:

Corollary

The Kruskal-Wallis test is consistent under stacking, as are any procedures (such as Mann-Whitney) which only rely on pairwise data.

Corollary

All tests derived from points-based voting procedures (such as the V test) are consistent under stacking of data sets with no Reversal component.

Corollary

Paradoxes due solely to Reversal components (for instance, including most differences between Kruskal-Wallis and the V test) lessen under stacking k times (and disappear in the limit as $k \to \infty$).

Proof of the Stacking Theorem

The proof is actually instructive and elegant. Recall the theorem:

Proof of the Stacking Theorem

The proof is actually instructive and elegant. Recall the theorem:

Theorem

If we stack an $n \times 3$ matrix of ranks k times, each Basic and Condorcet component is multiplied by k^2, each Reversal component is multiplied by k, and the Kernel component is multiplied by k^3.

Proof of the Stacking Theorem

The proof is actually instructive and elegant. Recall the theorem:

Theorem

If we stack an $n \times 3$ matrix of ranks k times, each Basic and Condorcet component is multiplied by k^2, each Reversal component is multiplied by k, and the Kernel component is multiplied by k^3.

(The proof of the Kernel may be done trivially. For a general $p \times 3$ matrix of ranks, there are p^3 triplets, so the size of the kernel is $p^3/6$; hence, for a $kp \times 3$ matrix, we get $k^3(p^3/6)$ as the size.)

Proof of the Stacking Theorem

The proof is actually instructive and elegant. Recall the theorem:

Theorem

If we stack an $n \times 3$ *matrix of ranks* k *times, each Basic and Condorcet component is multiplied by* k^2*, each Reversal component is multiplied by* k*, and the Kernel component is multiplied by* k^3*.*

The rest of the proof comes down to two lemmas:

Proof of the Stacking Theorem

The proof is actually instructive and elegant. Recall the theorem:

Theorem

If we stack an $n \times 3$ matrix of ranks k times, each Basic and Condorcet component is multiplied by k^2, each Reversal component is multiplied by k, and the Kernel component is multiplied by k^3.

The rest of the proof comes down to two lemmas:

Lemma

All triplets that are formed from elements taken from three different stanzas add only kernel components to the resulting profile decomposition.

Proof of the Stacking Theorem

The proof is actually instructive and elegant. Recall the theorem:

Theorem

If we stack an $n \times 3$ matrix of ranks k times, each Basic and Condorcet component is multiplied by k^2, each Reversal component is multiplied by k, and the Kernel component is multiplied by k^3.

The rest of the proof comes down to two lemmas:

Lemma

All triplets that are formed from elements taken from three different stanzas add only kernel components to the resulting profile decomposition.

(In fact, for $m > 3$ 'candidates', all m-tuplets formed from elements taken from m different stanzas add only kernel components.)

Proof of the Stacking Theorem

The proof is actually instructive and elegant. Recall the theorem:

Theorem

If we stack an $n \times 3$ matrix of ranks k times, each Basic and Condorcet component is multiplied by k^2, each Reversal component is multiplied by k, and the Kernel component is multiplied by k^3.

The rest of the proof comes down to two lemmas:

Lemma

All triplets that are formed from elements taken from three different stanzas add only kernel components to the resulting profile decomposition.

Lemma

For a stacking with $k = 2$, the Basic and Condorcet components are quadrupled, and each Reversal component is doubled.

Proof of the Stacking Theorem (cont.)

Lemma

Triplets from elements taken from three different stanzas add only kernel components.

Proof of the Stacking Theorem (cont.)

Lemma

Triplets from elements taken from three different stanzas add only kernel components.

One proves this by simply checking how many there are of each preference $X \succ Y \succ Z$, and it turns out there are exactly $\binom{k}{3} n^3$ of each.

Proof of the Stacking Theorem (cont.)

Lemma

Triplets from elements taken from three different stanzas add only kernel components.

Lemma

For $k = 2$, the Basic and Condorcet components are quadrupled, and the Reversal component is doubled.

Proof of the Stacking Theorem (cont.)

Lemma

Triplets from elements taken from three different stanzas add only kernel components.

Lemma

For $k = 2$, the Basic and Condorcet components are quadrupled, and the Reversal component is doubled.

One proves this by computing carefully how the initial profile vector (a, b, c, d, e, f) changes upon doubling (stacking $k = 2$), which is

$$(4a + b + c + e + f, a + 4b + c + d + f, a + b + 4c + d + e,$$
$$b + c + 4d + e + f, a + c + d + 4e + f, a + b + d + e + 4f).$$

Now multiplying both of these profiles by the decomposition matrix and comparing the two results yields the lemma.

Proof of the Stacking Theorem (cont.)

Now we prove the theorem.

Lemma

Triplets from elements taken from three different stanzas add only kernel components.

Lemma

For $k = 2$, the Basic and Condorcet components are quadrupled, and the Reversal component is doubled.

Proof of the Stacking Theorem (cont.)

Now we prove the theorem.

Lemma

Triplets from elements taken from three different stanzas add only kernel components.

Lemma

For $k = 2$, the Basic and Condorcet components are quadrupled, and the Reversal component is doubled.

Considering the k stanzas individually, we get k times the original components.
(So the second lemma really is just saying that when $k = 2$, we get no additional Reversal, but double our Basic and Condorcet.)

Proof of the Stacking Theorem (cont.)

Now we prove the theorem.

Lemma

Triplets from elements taken from three different stanzas add only kernel components.

Lemma

For $k = 2$, the Basic and Condorcet components are quadrupled, and the Reversal component is doubled.

Considering the k stanzas individually, we get k times the original components.

The first lemma indicates we only need to look at rankings coming from two different stanzas, of which there are $\binom{k}{2}$ possible choices. So we obtain $2\binom{k}{2} = k^2 - k$ additional (B_X and C, but **not** R_X) components.

Adding these to the k components we already have gives k^2, as desired, except for Reversal which remains at k, also as desired.

Outline

First Results

- The results so far lead one to ask about the component which behaves best in terms of paradoxes, and what results we might have regarding that. This is of course the Basic component.

First Results

- The results so far lead one to ask about the component which behaves best in terms of paradoxes, and what results we might have regarding that. This is of course the Basic component.
- Although there is no set which has only a Basic component (nor any profile with a positive number of voters!), we call any voting profile with only Kernel and Basic non-vanishing components a *Pure Basic*.

First Results

- The results so far lead one to ask about the component which behaves best in terms of paradoxes, and what results we might have regarding that. This is of course the Basic component.
- Although there is no set which has only a Basic component (nor any profile with a positive number of voters!), we call any voting profile with only Kernel and Basic non-vanishing components a *Pure Basic*.
- Hence the following results are useful!

First Results

- The results so far lead one to ask about the component which behaves best in terms of paradoxes, and what results we might have regarding that. This is of course the Basic component.
- Although there is no set which has only a Basic component (nor any profile with a positive number of voters!), we call any voting profile with only Kernel and Basic non-vanishing components a *Pure Basic*.
- Hence the following results are useful!

Theorem

Stacking can yield matrices of ranks with as large a Basic component as one desires, without being pure Basic.

Fact

Pure Basic data sets exist.

Proofs of First Results

Theorem

Stacking can yield matrices of ranks with as large a Basic component as one desires, without being pure Basic.

Proofs of First Results

Theorem

Stacking can yield matrices of ranks with as large a Basic component as one desires, without being pure Basic.

Take any matrix with no Condorcet component. Now just note that Pk^2 eventually outstrips Qk, no matter what P, Q are.

Proofs of First Results

Theorem

Stacking can yield matrices of ranks with as large a Basic component as one desires, without being pure Basic.

Fact

Pure Basic data sets exist.

Proofs of First Results

Theorem

Stacking can yield matrices of ranks with as large a Basic component as one desires, without being pure Basic.

Fact

Pure Basic data sets exist.

Implicit in Bargagliotti and Saari (2007) are propositions that if a profile comes from a pure Basic data set, it must have n^3 divisible by both 2 and 3, and hence n is divisible by six. Now a direct computation using the open source mathematics software Sage revealed that out of over seventeen million possible data sets of size $n = 6$, only about eight thousand were pure Basic - but they were there!

Proofs of First Results

Theorem

Stacking can yield matrices of ranks with as large a Basic component as one desires, without being pure Basic.

Fact

Pure Basic data sets exist.

Implicit in Bargagliotti and Saari (2007) are propositions that if a profile comes from a pure Basic data set, it must have n^3 divisible by both 2 and 3, and hence n is divisible by six. Now a direct computation using the open source mathematics software Sage revealed that out of over seventeen million possible data sets of size $n = 6$, only about eight thousand were pure Basic - but they were there!
See also the relevant note in the *Communications of the ACM*. Note that we still need the theorems, since the next possible size ($n = 12$) is approximately nine orders of magnitude more difficult of a computation!

Characterizing Pure Basics

We are nowhere near a full characterization of pure Basic data sets, not even at the level of the characterizations of pure Condorcet, Reversal, and Kernel voting profiles arising from nonparametric data sets found in Bargagliotti and Saari (2007). Nonetheless, there are interesting first steps.

Characterizing Pure Basics

We are nowhere near a full characterization of pure Basic data sets, not even at the level of the characterizations of pure Condorcet, Reversal, and Kernel voting profiles arising from nonparametric data sets found in Bargagliotti and Saari (2007). Nonetheless, there are interesting first steps.

Theorem

If any three entries in a pure Basic profile vector are known, or if we know two entries which do not correspond to opposite rankings (such as $A \succ B \succ C$ and $C \succ B \succ A$), it is possible to find the remaining entries.

Characterizing Pure Basics

We are nowhere near a full characterization of pure Basic data sets, not even at the level of the characterizations of pure Condorcet, Reversal, and Kernel voting profiles arising from nonparametric data sets found in Bargagliotti and Saari (2007). Nonetheless, there are interesting first steps.

Theorem

If any three entries in a pure Basic profile vector are known, or if we know two entries which do not correspond to opposite rankings (such as $A \succ B \succ C$ and $C \succ B \succ A$), it is possible to find the remaining entries.

Theorem

If $n = 6\ell$ is the size of the data set and the data set is pure Basic, then all entries in the underlying profile vector are divisible by 3ℓ.

Characterizing Pure Basics

We are nowhere near a full characterization of pure Basic data sets, not even at the level of the characterizations of pure Condorcet, Reversal, and Kernel voting profiles arising from nonparametric data sets found in Bargagliotti and Saari (2007). Nonetheless, there are interesting first steps.

Theorem

If any three entries in a pure Basic profile vector are known, or if we know two entries which do not correspond to opposite rankings (such as $A \succ B \succ C$ and $C \succ B \succ A$), it is possible to find the remaining entries.

Theorem

If $n = 6\ell$ is the size of the data set and the data set is pure Basic, then all entries in the underlying profile vector are divisible by 3ℓ.

For instance, all profile entries from a pure Basic data set with six observations are divisible by three. These are the first results we know of along these lines, which rely in a fundamental way upon the profile arising from a nonparametric data set.

Proving Characterizations

Theorem

If any three entries in a pure Basic profile vector are known, or if we know two entries which do not correspond to reversed rankings (such as $A \succ B \succ C$ *and* $C \succ B \succ A$*), it is possible to find the remaining entries.*

Proving Characterizations

Theorem

If any three entries in a pure Basic profile vector are known, or if we know two entries which do not correspond to reversed rankings (such as $A \succ B \succ C$ and $C \succ B \succ A$), it is possible to find the remaining entries.

For three, the proof is simply linear algebra. For two, it is in addition necessary to use the proofs of the lemmas from earlier which guarantee that n is divisible by 2 and 3. There **do** exist non-equivalent pure Basic profiles where two reversed rankings have the same numbers in the profile, so this theorem is sharp.

Proving Characterizations

Theorem

If any three entries in a pure Basic profile vector are known, or if we know two entries which do not correspond to reversed rankings (such as $A \succ B \succ C$ and $C \succ B \succ A$), it is possible to find the remaining entries.

Theorem

If $n = 6\ell$ is the size of the data set and the data set is pure Basic, then all entries in the underlying profile vector are divisible by 3ℓ.

Proving Characterizations

Theorem

If any three entries in a pure Basic profile vector are known, or if we know two entries which do not correspond to reversed rankings (such as $A \succ B \succ C$ and $C \succ B \succ A$), it is possible to find the remaining entries.

Theorem

If $n = 6\ell$ is the size of the data set and the data set is pure Basic, then all entries in the underlying profile vector are divisible by 3ℓ.

In fact, if one decomposes a profile coming from a nonparametric data set with n rows, one can prove that the Basic components are all multiples of $n/6$, the Reversal components are either multiples of $1/3$ or $1/6$, and the Condorcet component is either an even or odd multiple of $n/6$! (These last two depend on whether n is even or odd.)

Proving Characterizations (cont.)

Theorem

If $n = 6\ell$ is the size of the data set and the data set is pure Basic, then all entries in the underlying profile vector are divisible by 3ℓ.

Proving Characterizations (cont.)

Theorem

If $n = 6\ell$ is the size of the data set and the data set is pure Basic, then all entries in the underlying profile vector are divisible by 3ℓ.

To prove this theorem, we need a new concept - that of a *transposition* or *swap* of two elements (i, j) of a matrix of ranks. This is simply a switch of these ranks between two matrices of ranks.

Proving Characterizations (cont.)

Theorem

If $n = 6\ell$ is the size of the data set and the data set is pure Basic, then all entries in the underlying profile vector are divisible by 3ℓ.

To prove this theorem, we need a new concept - that of a *transposition* or *swap* of two elements (i, j) of a matrix of ranks. This is simply a switch of these ranks between two matrices of ranks.

The following shows a $(5, 2)$ transposition:

$$\begin{pmatrix} 6 & 5 & 4 \\ 1 & 3 & 2 \end{pmatrix} \text{ becomes } \begin{pmatrix} 6 & 3 & 5 \\ 1 & 2 & 4 \end{pmatrix}.$$

Proving Characterizations (cont.)

Theorem

If $n = 6\ell$ is the size of the data set and the data set is pure Basic, then all entries in the underlying profile vector are divisible by 3ℓ.

To prove this theorem, we need a new concept - that of a *transposition* or *swap* of two elements (i, j) of a matrix of ranks. This is simply a switch of these ranks between two matrices of ranks.

The set of all neighbor swaps $(i, i-1)$ from a given matrix of ranks will generate *all* possible matrices of ranks for a given shape $n \times 3$. In particular, we can begin with a canonical 'unanimity' matrix of ranks which has profile $(n^3, 0, 0, 0, 0, 0)$ and decomposition $\frac{n^3}{6}(B_A - B_C - R_B + C + K)$ and work from this fixed point.

Proving Characterizations (cont.)

Theorem

If $n = 6\ell$ is the size of the data set and the data set is pure Basic, then all entries in the underlying profile vector are divisible by 3ℓ.

To prove this theorem, we need a new concept - that of a *transposition* or *swap* of two elements (i, j) of a matrix of ranks. This is simply a switch of these ranks between two matrices of ranks.

The set of all neighbor swaps $(i, i-1)$ from a given matrix of ranks will generate *all* possible matrices of ranks for a given shape $n \times 3$. In particular, we can begin with a canonical 'unanimity' matrix of ranks which has profile $(n^3, 0, 0, 0, 0, 0)$ and decomposition $\frac{n^3}{6}(B_A - B_C - R_B + C + K)$ and work from this fixed point.

Finally, since n must be even, we let $n = 2k$ and write the decomposition as $\frac{4k^3}{3}(B_A - B_C - R_B + C + K)$.

Proving Characterizations (cont.)

Now we can outline the proof.

Proving Characterizations (cont.)

Now we can outline the proof.

Lemma

Any neighbor transposition $(i, i-1)$ *between the columns for candidates* Y *and* Z *(respectively) changes the Condorcet component by* $\pm\frac{2k}{3}$, *the Basic component by* $\frac{k}{3}(B_Z - B_Y)$, *and the Reversal component by an integer multiple of* $\frac{1}{6}(R_Y - R_Z)$.

Lemma

A sequence of neighbor transpositions which brings the Condorcet component to zero makes the Basic component an integer multiple of k.

Proving Characterizations (cont.)

Now we can outline the proof.

Lemma

Any neighbor transposition $(i, i-1)$ between the columns for candidates Y and Z (respectively) changes the Condorcet component by $\pm\frac{2k}{3}$, the Basic component by $\frac{k}{3}(B_Z - B_Y)$, and the Reversal component by an integer multiple of $\frac{1}{6}(R_Y - R_Z)$.

Lemma

A sequence of neighbor transpositions which brings the Condorcet component to zero makes the Basic component an integer multiple of k.

The proofs of the lemmas are unenlightening computations with voting profile differentials, and we omit them here.

Proving Characterizations (cont.)

Now we can outline the proof.

Lemma

Any neighbor transposition $(i, i-1)$ *between the columns for candidates* Y *and* Z *(respectively) changes the Condorcet component by* $\pm\frac{2k}{3}$*, the Basic component by* $\frac{k}{3}(B_Z - B_Y)$*, and the Reversal component by an integer multiple of* $\frac{1}{6}(R_Y - R_Z)$*.*

Lemma

A sequence of neighbor transpositions which brings the Condorcet component to zero makes the Basic component an integer multiple of k*.*

Proof of Theorem.

Recall that if $n = 6\ell$, then $k = 3\ell$, so that the Basic components are a multiple of 3ℓ. The Kernel also is, as $n^3/6 = (6\ell)(6\ell)(2k)/6 = 3\ell(4k\ell)$, and clearly the Condorcet and Reversal components are, since they are zero! Then we multiply by the (integer!) column matrix obtained from the basis, whereupon all entries are still divisible by 3ℓ. □

Outline

Directions to Proceed

There is of course plenty more work to do in this regard!

Questions

Directions to Proceed

There is of course plenty more work to do in this regard!

Questions

- *Will stacking help us with other aggregation questions?*
- *Can one say more about aggregation directly from the raw matrix of ranks (in the vein of Haunsperger or Bargagliotti), and not just using the proxy of voting profiles?*

Directions to Proceed

There is of course plenty more work to do in this regard!

Questions

- *Will stacking help us with other aggregation questions?*
- *Can one say more about aggregation directly from the raw matrix of ranks (in the vein of Haunsperger or Bargagliotti), and not just using the proxy of voting profiles?*
- *On a somewhat more ambitious note, one could also try to generalize the specifics of some of these ideas for* $n > 3$. *This seems harder.*

Directions to Proceed

There is of course plenty more work to do in this regard!

Questions

- *Will stacking help us with other aggregation questions?*
- *Can one say more about aggregation directly from the raw matrix of ranks (in the vein of Haunsperger or Bargagliotti), and not just using the proxy of voting profiles?*
- *On a somewhat more ambitious note, one could also try to generalize the specifics of some of these ideas for* $n > 3$*. This seems harder.*
- *On a very ambitious note, can one characterize the subset of general voting profile space that matrices of ranks generate?*

Acknowledgments

Finally, I'd like to thank the following:

Acknowledgments

Finally, I'd like to thank the following:

- Sarah Berube - for her enthusiasm and talent as a research and REU student, and collaborator

Acknowledgments

Finally, I'd like to thank the following:

- Sarah Berube - for her enthusiasm and talent as a research and REU student, and collaborator
- Anna Bargagliotti - for helpful emails and encouraging the project

Acknowledgments

Finally, I'd like to thank the following:

- Sarah Berube - for her enthusiasm and talent as a research and REU student, and collaborator
- Anna Bargagliotti - for helpful emails and encouraging the project
- The Gordon College Faculty Development Committee - for the Initiative Grant which made the REU possible

Acknowledgments

Finally, I'd like to thank the following:

- Sarah Berube - for her enthusiasm and talent as a research and REU student, and collaborator
- Anna Bargagliotti - for helpful emails and encouraging the project
- The Gordon College Faculty Development Committee - for the Initiative Grant which made the REU possible
- Mike Veatch and the queuing theory group at Gordon - for a good work environment

Acknowledgments

Finally, I'd like to thank the following:

- Sarah Berube - for her enthusiasm and talent as a research and REU student, and collaborator
- Anna Bargagliotti - for helpful emails and encouraging the project
- The Gordon College Faculty Development Committee - for the Initiative Grant which made the REU possible
- Mike Veatch and the queuing theory group at Gordon - for a good work environment
- Don Saari - for organizing this conference and helpful feedback

Acknowledgments

Finally, I'd like to thank the following:

- Sarah Berube - for her enthusiasm and talent as a research and REU student, and collaborator
- Anna Bargagliotti - for helpful emails and encouraging the project
- The Gordon College Faculty Development Committee - for the Initiative Grant which made the REU possible
- Mike Veatch and the queuing theory group at Gordon - for a good work environment
- Don Saari - for organizing this conference and helpful feedback
- All of you - for coming!