# SPATIAL MODELS OF PARLIAMENTARY VOTING

**By**

**Keith T. Poole**

## Chapter 3: The Optimal Classification Method

### *Overview*

In this chapter I show a method – Optimal Classification (OC) – that is based upon the geometry shown in Chapter 2 and is designed to analyze real-world roll call data. Simply put, I add error to the simple spatial model developed in Chapter 2 and show an estimation method – Optimal Classification (OC) -- that *maximizes the correct classification* of legislative choices. In Chapter 4 I will focus entirely on models that *maximize the probabilities* of legislative choices using the same error framework. I show OC first because it can be used as a very reliable platform upon which to build more intricate estimation methods.

Within the psychometrics field, the OC scaling method is a *nonmetric unfolding* procedure. It is an "unfolding" because the roll calls are treated as preferential choice data and parameters for individuals (legislators) and stimuli (roll calls) are being estimated. It is "nonmetric" because no assumptions are made about the parametric form of the individuals' "true" preference functions other than that they are symmetric and single peaked.

Unfolding was developed for the one-dimensional case by Coombs (1950) and generalized to the multidimensional case by Bennett and Hays (1960). The original unfolding model – later dubbed the *ideal-point model* – represented individuals and stimuli as points and was originally developed to analyze rank orderings of stimuli by individuals. Later, Tucker (1960) developed the *vector model* of unfolding in which the individuals are treated as vectors and the stimuli as points. The vector model is a special case of the unfolding model where the individual's ideal point goes off to infinity (Carroll, 1980; Borg and Groenen, 1997, ch. 15). This model is much like Guttman scaling in that the individual's utility rises/falls monotonically from the center of the space off to infinity along the individual's vector. The projections of the stimuli onto the individual's vector reproduce the observed rank ordering. The vector model is the basis of the MDPREF program developed by Chang and Carroll (1969).

With respect to the roll call voting problem, the ideal point and vector unfolding models are closely related. If the individuals are treated as roll calls and the roll calls are treated as individuals then the individual becomes a cutting plane through the space and the point where the cutting plane passes through the normal (individual) vector is the *individual's threshold*. That is, the individual approves/accepts the stimuli on one side of the plane and disapproves/does not accept the stimuli on the other side of the plane. "Pick Any/N" data that are widely used in marketing applications (DeSarbo and Cho, 1989) have this form. For example, respondents are given a list of soda pops and asked if they drink/not drink each soda. The soda pops are then displayed as points in a space and the individuals as cutting lines that divide the soda pops into drink/not drink.

Psychometricians have largely abandoned the nonmetric approach in the past fifteen years "because they suspected instability and identification problems" and have focused their efforts on probabilistic and metric alternatives.[1] The most important recent work along these lines on the roll call voting problem is by Londregan (2000).

Londregan links the psychometrics testing literature with the spatial theory of legislative voting and derives important statistical results about the parameters of the spatial model. I will discuss Londregan's work in Chapter 4.

At the heart of OC are two algorithms -- *the cutting plane procedure* and *the legislative procedure*. Both of these procedures are unique and are very stable. In particular, when the number of legislators is 100 or greater and the number of roll calls is on the order of 500 – typical of national legislatures, for example, the U.S. Senate – then the recovery of the legislators and cutting lines/planes is very precise. With 500 roll calls, there are a maximum 125,251 polytopes in two dimensions and a maximum of 20,833,751 polytopes in three dimensions. Most of these polytopes are so small that a typical legislator's point is very precisely pinned down. In fact the recovery of the legislator coordinates is virtually identical to those recovered by parametric procedures that must make strong assumptions about the interpersonal comparability of individuals' utility and the function form of the error distribution (e.g., Heckman and Snyder, 1997; Poole and Rosenthal, 1997). In addition, at very low levels of error,[2] OC is stable and is not susceptible to the problems encountered with many parametric procedures (Rosenthal and Voeten, 2004).

To recap the notation used in Chapter 2, each legislator is represented by an ideal point, a s by 1 vector $\mathbf{X_i}$, in the policy space with a symmetric single-peaked utility function centered at the ideal point (see Figure 2.1) and each roll call is represented as two points – one for the "Yea" policy outcome ($\mathbf{O_{jy}}$) and one for the "Nay" policy outcome ($\mathbf{O_{jn}}$) (both s by 1 vectors). If there were no error the legislator would vote *deterministically* – that is, she would *always* vote for the closest alternative in the policy space. When *random* error is added to the utility function then the legislator votes *probabilistically* – that is, depending upon the random error, sometimes she votes for the closest alternative and sometimes she does not.

I introduce error into a legislator's choice process by using the *random utility model* (McFadden, 1976). In the random utility model a legislator's overall utility for Yea/Nay is the sum of a deterministic utility and a random error. Suppose there are p legislators, q roll calls, and s dimensions indexed by i=1,..,p, j=1,...,q, and k=1,…,s, respectively. Legislator i's utility for the Yea outcome on roll call j is:

$$U_{ijy} = u_{ijy} + \varepsilon_{ijy} \qquad (3.1)$$

where $u_{ijy}$ is the deterministic portion of the utility function and $\varepsilon_{ijy}$ is the stochastic or random portion of the utility function. In Chapter 4 I will discuss probabilistic voting models that are built up from assumptions about the functional forms of the deterministic and stochastic portions of $U_{ijy}$. In the remainder of this chapter I assume only that legislators utilize a *symmetric single-peaked* utility function and discuss the general problems of (1) estimating a roll call cutting plane that maximizes correct classification given the legislator ideal points; and (2) estimating a legislator ideal point that maximizes correct classification given the cutting planes.

I discuss the one-dimensional maximum classification problem first because the same algorithm can be used to estimate *both* roll call cutting points and legislator ideal points. In Chapter 2 I show the one-dimensional solution for the roll call voting problem when there is no error. Consequently, in my discussion of the one-dimensional maximum classification algorithm I will assume that error is present. In the multidimensional problem, there is (as yet) no solution for the perfect voting problem.

However, the cutting plane procedure and legislator procedure that are used by OC in two or more dimensions do an excellent job of solving both the perfect voting problem and the roll call voting with error.

### *The One Dimensional Maximum Classification Scaling Problem – The "Janice" Algorithm*

In Chapter 2 I show a solution to the problem of one-dimensional perfect voting; namely, to get the legislator coordinates:

1. Compute the p by p agreement score matrix
2. Convert the agreement score matrix into a matrix of squared distances
3. Double Center the matrix of squared distances
4. Take the square root of a diagonal element of the double centered matrix and divide it through the corresponding column

In the appendix to Chapter 2 I show that the first eigenvector extracted from the double-centered transformed agreement score matrix has the same rank ordering as the true data. This is equivalent to step (4) above. When error is present this method cannot *guarantee* that the rank ordering of the first eigenvector maximizes correct classification. However, the first eigenvector provides excellent *starting values* for a rank order that can be iteratively improved by what I call *the Janice Algorithm* to arrive at a rank ordering that almost certainly maximizes correct classification. I will first discuss the case of very low error and then the more general case of higher error.

### The Effect of Very Low Error in One Dimension

Suppose the error, $\varepsilon_{ijy}$, is drawn from some continuous probability distribution with mean $\mu$ and variance $\sigma^2$; that is,

$$\varepsilon_{ijy} \sim f(\mu,\sigma^2)$$

Suppose σ is very small, that is, suppose σ is an *infinitesimal quantity of the first order*, "that is, a quantity whose higher powers are, for the problem at hand, negligible in comparison to lower powers of [σ]" (Courant and Hilbert, 1937, vol. 1, p. 41), then on a roll call only the legislators adjacent and a very tiny distance away from a cutting point will make a voting error. However, this is observationally indistinguishable from perfect voting! This is shown in Figures 3.1, 3.2, and 3.3.

Figure 3.1 is the same as Figure 2.2 – Perfect voting by six legislators on five roll calls with the Yea alternative always on the left. The top part of the figure shows the voting along the dimension and the lower part of the figure shows the voting in the form of a roll call matrix. The lower triangle of the roll call matrix is all Nays and the upper triangle is all Yeas. Now, suppose legislators Three and Four are very close together with the cutting point for the third roll call between them. With very low error, either, neither, or both legislators could make a voting error. Figure 3.2 illustrates the four possibilities.

### Figure 3.1 Perfect Spatial Voting in One Dimension

-1---------0--------+1

Legislators $X_1$ $X_2$ $X_3$ $X_4$ $X_5$ $X_6$

Cutpoints $Z_1$ $Z_2$ $Z_3$ $Z_4$ $Z_5$

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | Y | N | N | N | N | N |
| 2 | Y | Y | N | N | N | N |
| 3 | Y | Y | Y | N | N | N |
| 4 | Y | Y | Y | Y | N | N |
| 5 | Y | Y | Y | Y | Y | N |

Roll Call Matrix

| | Roll Calls | | | | |
|---|---|---|---|---|---|
| Legislators | 1 | 2 | 3 | 4 | 5 |
| One | Y | Y | Y | Y | Y |
| Two | N | Y | Y | Y | Y |
| Three | N | N | Y | Y | Y |
| Four | N | N | N | Y | Y |
| Five | N | N | N | N | Y |
| Six | N | N | N | N | N |

The first possibility, case (a), shown in the first row of Figure 3.2, is that neither legislator makes an error; that is, for legislator Three $U_{33y} > U_{33n}$ so she votes Yea even with the error, and for legislator Four $U_{43n} > U_{43y}$ so he votes Nay even with the error. In case (b) the addition of the error causes legislator Three's Utility for Nay to become larger than her utility for Yea -- $U_{33y} < U_{33n}$ -- so she now votes Nay and legislator Four still votes Nay even with the error. Note that this produces a roll call vote that is identical to the second roll call in Figure 3.1.

Similarly, in case (c), the addition of the error changes Legislator Four's vote to Yea while the error does not affect Legislator Three's vote. Note that this produces a roll call vote that is identical to the fourth roll call in Figure 3.1.

Finally, in case (d), the addition of the error changes the votes of both legislators.

**Figure 3.2 Low Noise Spatial Voting in One Dimension On Third Roll Call**

| Legislators | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | $X_6$ | |
|---|---|---|---|---|---|---|---|
| | | | | $Z_3$ | | | |
| Low Noise (a) | Y | Y | Y | N | N | N | |
| Low Noise (b) | Y | Y | N | N | N | N | (same as RC 2) |
| Low Noise (c) | Y | Y | Y | Y | N | N | (same as RC 4) |
| Low Noise (d) | Y | Y | N | Y | N | N | |

To recap, cases (a), (b), and (c) are identical to roll call votes 3, 2, and 4, respectively. Consequently, the presence of the error would be unknowable. In case (a) nothing changes. In case (b) it would look like the cutting points for roll calls 2 and 3 were the same and in case (c) it would look like the cutting points for roll calls 3 and 4 were the same.

This leaves case (d). However, as Figure 3.3 shows, the presence of the error even in this “obvious” case also would be unknowable because legislator Three’s voting pattern is now identical to Four’s in Figure 3.1. That is, simply transposing the rows corresponding to legislators Three and Four as shown in Figure 3.3 produces a perfect voting pattern.

Figure 3.3 Roll Call Matrix With Low Noise

| | Roll Calls | | | | |
|---|---|---|---|---|---|
| Legislators | 1 | 2 | 3d | 4 | 5 |
| One | Y | Y | Y | Y | Y |
| Two | N | Y | Y | Y | Y |
| Three | N | N | N | Y | Y |
| Four | N | N | Y | Y | Y |
| Five | N | N | N | N | Y |
| Six | N | N | N | N | N |

Perfect Voting With Four and Three Transposed

| | Roll Calls | | | | |
|---|---|---|---|---|---|
| Legislators | 1 | 2 | 3d | 4 | 5 |
| One | Y | Y | Y | Y | Y |
| Two | N | Y | Y | Y | Y |
| Four | N | N | Y | Y | Y |
| Three | N | N | N | Y | Y |
| Five | N | N | N | N | Y |
| Six | N | N | N | N | N |

In sum, when the error level is very low *it cannot be detected*! The effect will be to perhaps slightly scramble the "true" ordering but we can never know whether or not it is occurring because it is observationally indistinguishable from perfect voting!

**The Effect of Higher Levels of Error in One Dimension**

Suppose the error is high enough so that rows and columns of the simple roll call matrix in Figure 3.1 cannot be swapped or transposed so as to produce a perfect voting matrix. In this instance, the error will be detected. An example of this is shown in Figures 3.4 and 3.5. In Figure 3.4 the only error is legislator One voting Nay on roll call 3 when she should vote Yea (shown in underline and italics). Using the four-step method discussed above, Figure 3.4 also shows the agreement score matrix and the first eigenvector extracted from the double-centered transformed agreement score matrix. The ordering of the legislators implied by the eigenvector is:

$$X_2 < X_1 < X_3 < X_4 < X_5 < X_6$$

So that the single voting error by legislator One has transposed the order of legislators One and Two. This rank ordering is used in Figure 3.5 to illustrate how the Janice algorithm works.

## Figure 3.4 Recovering the Legislators

## With Higher Error

| | Roll Calls | | | | |
|---|---|---|---|---|---|
| Legislators | 1 | 2 | 3 | 4 | 5 |
| One | Y | Y | *N* | Y | Y |
| Two | N | Y | Y | Y | Y |
| Three | N | N | Y | Y | Y |
| Four | N | N | N | Y | Y |
| Five | N | N | N | N | Y |
| Six | N | N | N | N | N |

**Agreement Scores**

| | | | | | |
|---|---|---|---|---|---|
| 1.0 | | | | | |
| .6 | 1.0 | | | | |
| .4 | .8 | 1.0 | | | |
| .6 | .6 | .8 | 1.0 | | |
| .4 | .4 | .6 | .8 | 1.0 | |
| .2 | .2 | .4 | .6 | .8 | 1.0 |

**First Eigenvector**

$X_1 = -.44512$

$X_2 = -.48973$

$X_3 = -.11093$

$X_4 = .04431$

$X_5 = .34859$

$X_6 = .65288$

Figure 3.5 shows how the Janice algorithm finds roll call cutting points that maximize correct classification given a rank ordering of legislators. With six legislators there are seven possible rank positions for a roll call cutting point. These are shown as rows in the Figure. For example, if the cutting point is placed before the left-most legislator, in this case $Z_j < X_2$ then this would imply if voting was perfect that all the legislators would vote for the alternative to the right of the cutting point – Nay in the upper part of the Figure and Yea in the lower part of the Figure. Similarly, if the cutting point is located between $X_2$ and $X_1$, that is, $X_2 < Z_j < X_1$, then with perfect voting with Yea as the left alternative legislator Two votes Yea and the other legislators vote Nay and vice versa if Yea is the right alternative. This is shown in the second row of both parts of Figure 3.5.

The columns on the right side of Figure 3.5 correspond to the five roll calls from Figure 3.4. The row entries in each column show the number of classification errors when the corresponding cutting point is used for that roll call. For example, for roll call 1, if the cutting point is placed furthest to the left, that is, $Z_1 < X_2$, then this results in only 1 classification error. In terms of the legislator ordering, the pattern of roll call 1 is NYNNNN so that predicting NNNNNN produces only one classification error – legislator One is predicted to vote Nay and she actually votes Yea. Similarly, if the cutting point is placed between Legislators Two and One, namely, $X_2 < Z_1 < X_1$, then the predicted pattern is YNNNNN. This produces two classification errors – both legislators Two and One are wrongly predicted. Placing the cutting point between legislators One and Three, $X_1 < Z_1 < X_3$, predicts the pattern YYNNNN which produces only one classification error – legislator Two. Note that moving up or down the column can only

# Figure 3.5 The Janice Algorithm: Roll Calls

**Legislators** $X_2 < X_1 < X_3 < X_4 < X_5 < X_6$

**Predicted Patterns of All Possible Rank Positions For a Roll Call Cutting point Given the Above Legislator Ordering**

**(Yea on the Left, Nay on the Right)**

| | | | | | | | Number of Errors On Roll Calls | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 1 | 2 | 3 | 4 | 5 |
| $Z_j < X_2$ | N | N | N | N | N | N | 1 | 2 | 2 | 4 | 5 |
| $X_2 < Z_j < X_1$ | Y | N | N | N | N | N | 2 | 1 | 1 | 3 | 4 |
| $X_1 < Z_j < X_3$ | Y | Y | N | N | N | N | *1* | *0* | 2 | 2 | 3 |
| $X_3 < Z_j < X_4$ | Y | Y | Y | N | N | N | 2 | 1 | *1* | 1 | 2 |
| $X_4 < Z_j < X_5$ | Y | Y | Y | Y | N | N | 3 | 2 | 2 | *0* | 1 |
| $X_5 < Z_j < X_6$ | Y | Y | Y | Y | Y | N | 4 | 3 | 3 | 1 | *0* |
| $X_6 < Z_j$ | Y | Y | Y | Y | Y | Y | 5 | 4 | 4 | 2 | 1 |

**(Nay on the Left, Yea on the Right)**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Z_j < X_2$ | Y | Y | Y | Y | Y | Y | 5 | 4 | 4 | 2 | 1 |
| $X_2 < Z_j < X_1$ | N | Y | Y | Y | Y | Y | 4 | 5 | 5 | 3 | 2 |
| $X_1 < Z_j < X_3$ | N | N | Y | Y | Y | Y | 5 | 6 | 4 | 4 | 3 |
| $X_3 < Z_j < X_4$ | N | N | N | Y | Y | Y | 4 | 5 | 5 | 5 | 4 |
| $X_4 < Z_j < X_5$ | N | N | N | N | Y | Y | 3 | 4 | 4 | 6 | 5 |
| $X_5 < Z_j < X_6$ | N | N | N | N | N | Y | 2 | 3 | 3 | 5 | 6 |
| $X_6 < Z_j$ | N | N | N | N | N | N | 1 | 2 | 2 | 4 | 5 |

increase or decrease the number of errors by one and that there can be multiple cutting points with the same number of errors.

The classification error cells shown in underline and italic correspond to the optimal locations of the corresponding cutting points. When there are two or more cutting points that produce the same best correct classification the rule is to pick the cutting point closest to the center of the legislator rank ordering. For example, for roll call 1, placing the cutting point furthest to the left produces one classification error when the Nay alternative is on the right. However, note that placing the cutting point furthest to the right, $X_6 < Z_1$ , also produces one classification error if the Yea alternative is on the right! This is a good illustration of the problems encountered with lop-sided roll calls in real world data. Oftentimes placing the cutting point at either end of the dimension with the polarity reversed in each case produces the same number of correct classifications! In both cases, the cutting point predicts a unanimous vote. However, if there are *interior* ranks that produce the same number of correct classifications the Janice algorithm always picks the most interior cutting point (that is, the one closest to the median rank).

The application of the Janice algorithm shown in Figure 3.5 produces the following joint ordering of legislators and cutting points:

$$X_2 < X_1 < Z_1 = Z_2 < X_3 < Z_3 < X_4 < Z_4 < X_5 < Z_5 < X_6$$

This joint ordering produces two classification errors – legislator Two is predicted to vote Yea on roll call 1 and he actually votes Nay, and legislator One is predicted to vote Yea on roll call 3 and she actually votes Nay.

Figure 3.6 shows how the Janice algorithm finds legislator rank positions that maximize correct classification given a rank ordering of roll call cutting points and the polarity – what alternative is on the left and what alternative is on the right -- of each roll call. These are shown as “L” and “R” in the figure. Because legislators do not have a “polarity” per se, Figure 3.6 is simpler than Figure 3.5. With five roll calls there are six possible rank positions for a legislator. These are shown as rows in the Figure. For example, if a legislator is placed before the left-most roll call cutting point, in this case $X_i < Z_1$, then with perfect voting this would imply that the legislator would vote for the Left alternative on every roll call – LLLLL in the first row of the figure. The fact that roll

**Figure 3.6 The Janice Algorithm: Legislators**

**Roll Calls** $Z_1 = Z_2 < Z_3 < Z_4 < Z_5$

**Predicted Patterns of All Possible Rank Positions For a Legislator Given the Above Cutting point Ordering**
**(Roll Call Polarity From Table 3.5)**

| | | | | | | Number of Errors On Legislators | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 1 | 2 | 3 | 4 | 5 | 6 |
| $X_i < Z_1$ | L | L | L | L | L | *1* | 1 | 2 | 3 | 4 | 5 |
| $Z_1 = X_i = Z_2$ | R | L | L | L | L | 2 | *0* | 1 | 2 | 3 | 4 |
| $Z_2 < X_i < Z_3$ | R | R | L | L | L | 3 | 1 | *0* | 1 | 2 | 3 |
| $Z_3 < X_i < Z_4$ | R | R | R | L | L | 2 | 2 | 1 | *0* | 1 | 2 |
| $Z_4 < X_i < Z_5$ | R | R | R | R | L | 3 | 3 | 2 | 1 | *0* | 1 |
| $Z_5 < X_i$ | R | R | R | R | R | 4 | 4 | 3 | 2 | 1 | *0* |

calls 1 and 2 are tied in terms of their rank does not present a problem *because of the fixed polarities*. Placing a legislator at the tied position -- $Z_1 = X_i = Z_2$ – would predict the patterns RLLLL or LRLLL. However, treating $Z_1$ as being "before" $Z_2$ or vice versa for bookkeeping purposes is not a problem. If the polarities of the two roll calls are the same at the tied position, then this also presents no problems in terms of the placement of a legislator rank. For example, if the polarity of roll calls 1 and 2 at the tied position was L, then the implied pattern would be LLLLL. However, this pattern is the same as the first row of the figure so that $X_i < Z_1$. If the polarity of both roll calls was R at the tied position, then the implied pattern would be RRLLL. This pattern is the same as the pattern produced by placing the legislator between the tied pair and roll call 3 – that is, $Z_1 = Z_2 < X_i < Z_3$ . In short, in all cases the placement of the legislator is unambiguous.

The columns on the right side of Figure 3.6 correspond to the six legislators. The row entries in each column show the number of classification errors when the corresponding legislator rank position vis a vis the roll call cutting points is used. For example, for legislator One, if the rank position is placed furthest to the left, that is, if $X_1 < Z_1$ , then this results in only 1 classification error. Legislator One votes YYNYY on roll calls 1 through 5. Given the polarities, this translates into the pattern LLRLL. Hence, predicting LLLLL by placing legislator One to the left of all the roll call cutting points only produces one classification error – she is predicted to vote L (or Yea) on roll call 3 when in fact she votes R (or Nay). Similarly, if legislator One's rank is placed at the tied rank position of roll calls 1 and 2, then the predicted pattern would be RLLLL and this produces two classification errors.

The classification error cells shown in underline and italic correspond to the optimal locations of the corresponding legislators. When there are two or more rank positions that produce the same best correct classification the rule is to pick the rank position for the legislator that is closest to the median of the roll call cutting point rank ordering.

The application of the Janice algorithm shown in Figure 3.6 produces the following joint ordering of legislators and cutting points:

$$X_1 < Z_1 = Z_2 = X_2 < X_3 < Z_3 < X_4 < Z_4 < X_5 < Z_5 < X_6$$

This joint ordering produces only one classification error – legislator One is predicted to vote Yea (L) on roll call 3 and she actually votes Nay (R).

Figure 3.7 shows the application of the Janice algorithm to the roll calls using the new rank ordering of the legislators. Figure 3.7 is laid out the same as Figure 3.5. The joint rank order produced by Figure 3.7 is:

$$X_1 < Z_1 < X_2 < Z_2 < X_3 < Z_3 < X_4 < Z_4 < X_5 < Z_5 < X_6$$

This joint ordering produces only one classification error – legislator One is predicted to vote Yea on roll call 3 and she actually votes Nay.

## Figure 3.7 The Janice Algorithm: Second Iteration For Roll Calls

**Legislators** $X_1 < X_2 < X_3 < X_4 < X_5 < X_6$

**Predicted Patterns of All Possible Rank Positions For a Roll Call Cutting point Given the Above Legislator Ordering**

**(Yea on the Left, Nay on the Right)**

| | | | | | | | Number of Errors On Roll Calls | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 1 | 2 | 3 | 4 | 5 |
| $Z_j < X_1$ | N | N | N | N | N | N | 1 | 2 | 2 | 4 | 5 |
| $X_1 < Z_j < X_2$ | Y | N | N | N | N | N | *0* | 1 | 3 | 3 | 4 |
| $X_2 < Z_j < X_3$ | Y | Y | N | N | N | N | 1 | *0* | 2 | 2 | 3 |
| $X_3 < Z_j < X_4$ | Y | Y | Y | N | N | N | 2 | 1 | *1* | 1 | 2 |
| $X_4 < Z_j < X_5$ | Y | Y | Y | Y | N | N | 3 | 2 | 2 | *0* | 1 |
| $X_5 < Z_j < X_6$ | Y | Y | Y | Y | Y | N | 4 | 3 | 3 | 1 | *0* |
| $X_6 < Z_j$ | Y | Y | Y | Y | Y | Y | 5 | 4 | 4 | 2 | 1 |

**(Nay on the Left, Yea on the Right)**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Z_j < X_1$ | Y | Y | Y | Y | Y | Y | 5 | 4 | 4 | 2 | 1 |
| $X_1 < Z_j < X_2$ | N | Y | Y | Y | Y | Y | 6 | 5 | 3 | 3 | 2 |
| $X_2 < Z_j < X_3$ | N | N | Y | Y | Y | Y | 5 | 6 | 4 | 4 | 3 |
| $X_3 < Z_j < X_4$ | N | N | N | Y | Y | Y | 4 | 5 | 5 | 5 | 4 |
| $X_4 < Z_j < X_5$ | N | N | N | N | Y | Y | 3 | 4 | 4 | 6 | 5 |
| $X_5 < Z_j < X_6$ | N | N | N | N | N | Y | 2 | 3 | 3 | 5 | 6 |
| $X_6 < Z_j$ | N | N | N | N | N | N | 1 | 2 | 2 | 4 | 5 |

Note that going from Figure 3.5 to Figure 3.6 and going from Figure 3.6 to Figure 3.7, *the classification error cannot increase*! This is due to the fact that, holding the rank ordering of the legislators/cutting points fixed, *the Janice algorithm always finds ranks for the cutting points/legislators that maximize correct classification because it checks all possible rank positions*. For example, in Figure 3.5 the starting estimate of the legislator rank ordering is used to get the first ordering of the cutting points. This first ordering of the cutting points is then used in Figure 3.6 to get a new legislator ordering. As shown in both Figures, the Janice algorithm always finds the ranks that maximize correct classification because it checks all possible rank positions. Hence, the new ordering of the legislators *must* classify as good as the initial ordering. A similar reasoning applies to the transition from Figure 3.6 to Figure 3.7.

An attractive feature of the Janice algorithm is its computational efficiency – namely, it is linear in the number of legislators/roll calls. In Figure 3.5 there are only 2p *unique* perfect patterns. Although, there are 2p+2 = 14 patterns shown in Figure 3.5, the unanimous patterns appear twice for purposes of clarity. Computationally, it is a simple matter to compare each perfect pattern with the actual pattern of votes. This can be done very efficiently by first assuming that the cutting point is to the left of all the legislators as in row one of the figure and then calculating the corresponding number of correct classifications. This requires p calculations. Next assume that the cutting point is between the leftmost pair of legislators – $X_2$ and $X_1$ in Figure 3.5. To get the number of correct classifications only one calculation has to be made because the only change is that the cutting point rank has been moved from the left of the leftmost legislator to between the leftmost pair of legislators. If there is no missing data, either the correct classification increases by 1 or decreases by 1 when the cutting point is moved one rank position in this fashion. For each possible cutting point rank the correct classification corresponding to the two possible perfect patterns can be calculated (the upper and lower portions of Figure 3.5). With no missing data, the number of correct classifications for a particular rank position for a perfect pattern and its mirror image must *always* add to p. Therefore, only 1 calculation is required when the cutting point is moved one rank position with no missing data to know the correct classifications for *both* polarities. Hence the total number of calculations required to find the maximum classification cutting point rank *and* its associated polarity is 2p.

The same reasoning holds for finding the legislator rank that maximizes correct classification given the cutting point ranks. Given the polarity of the roll calls as shown in Figure 3.6, only 2q calculations are required to find the maximum classification rank for each legislator.

In sum, the one-dimensional Optimal Classification method is:

1) Generate starting estimate of the legislator rank ordering
2) Holding the legislator rank ordering fixed, use the Janice algorithm to find the optimal cutting point ordering
3) Holding the cutting point ordering fixed, use the Janice algorithm to find the optimal legislator ordering
4) Go to (2)

Hereafter, I will refer to steps (2) to (4) as the *Edith Algorithm*.

This simple algorithm converges very rapidly to a solution in which the rank ordering of the legislators and the rank ordering of the roll call midpoints reproduce each other. In the example above,

Step (1) $X_2 < X_1 < X_3 < X_4 < X_5 < X_6$

Step (2a) $Z_1 = Z_2 < Z_3 < Z_4 < Z_5$

Step (3a) $X_1 < X_2 < X_3 < X_4 < X_5 < X_6$

Step (2b) $Z_1 < Z_2 < Z_3 < Z_4 < Z_5$

Step (3b) $X_1 < X_2 < X_3 < X_4 < X_5 < X_6$

Step (2c) $Z_1 < Z_2 < Z_3 < Z_4 < Z_5$

Etc etc

The Edith algorithm always converges to a solution in which the two rank orderings reproduce each other. This joint rank ordering of cutting points and legislators is a very strong form of *conditional global maximum*. Technically, if there are multiple sets of parameters (for example, as in this case, parameters corresponding to rows of a matrix and parameters corresponding to columns of a matrix), and every set of parameters is at a global maximum conditioned on the other sets being held fixed, and these sets reproduce each other, then they are at a conditional global maximum. Note that the overall global maximum, by definition, is a conditional global maximum. The global maximum in principle can be found by checking every p!/2 unique ordering of the legislators (the division by two removes the mirror image orderings). That is, for every unique ordering of the legislators, use the Janice algorithm to find the optimal rank positions of the cutting points. For small voting bodies of 15 or less, this could actually be done. Future massively parallel supercomputers could possibly bring into reach legislatures the size of the U.S. Senate (p=100) and U.S. House (p=435). Until then, the Edith algorithm will have to do!

However, an encouraging aspect of the Edith algorithm is that Conditional Global maxima are quite rare because of the nature of the constraints. Indeed, in the metric similarities problem, conditional global minima are very rare and *their number declines as the number of parameters increases* (Poole, 1990).

### *The Multidimensional Maximum Classification Scaling Problem*

In two or more dimensions there is as yet no solution for the perfect voting problem. Indeed, it is difficult to tell whether or not the data is even perfect! What made the one-dimensional perfect voting problem solvable was the fact that the number of cutting points between a pair of legislators could be converted into a Euclidean distance. In more than one dimension, legislators are identified only up to polytopes and the distances between these polytopes are not *necessarily* linear in the number of cutting lines/planes between them (see Figure 2.10). A consequence of this is that error, *which is manifest only in the changes of Yeas to Nays and Nays to Yeas in a legislator's voting pattern*, cannot be guaranteed to be *symmetric with direction* in the space.

Although these are serious problems, because the Optimal Classification method is built directly upon the underlying geometry, it can deal with them effectively. Although OC cannot be *guaranteed* to *always* find the best possible solution, it will come extremely close in almost all situations.

In the next section I will show how to estimate a cutting plane that maximizes correct classification on a roll call *given* an estimate of the legislator points. In the following section I will show how to find the polytope that maximizes the correct classification of a legislator's choices *given* an estimate of the roll call cutting planes.

## Estimating a Roll Call Cutting Plane Given the Legislator Ideal Points

Given the legislator ideal points – the p by s matrix **X** – and the choices of the legislators on a roll call, the problem is to find a line in two dimensions or a plane in more than two dimensions that divides the legislators into two groups so as to maximize the number of correct classifications. Figure 3.8 shows the basics of the problem in two dimensions.

Panel A of Figure 3.8 is the same as Figure 2.11 and shows the basic geometry of a roll call vote in two dimensions. The normal vector is denoted as $\mathbf{N_j}$ and its *reflection* as $-\mathbf{N_j}$. I impose the constraint that the normal vectors be of unit length, namely;

$$\mathbf{N_j}'\mathbf{N_j} = \sum_{k=1}^{s} N_{jk}^2 = 1 \qquad (3.2)$$

This constraint does not affect the fundamental geometry outlined in Chapter Two. I impose the constraint simply because unit length normal vectors make the algebra of the problem easier to explain. For example, the normal vector in Figure 3.8 is $\begin{bmatrix} \mathbf{.6} \\ \mathbf{.8} \end{bmatrix}$.

I will refer to the line formed by the normal vector and its reflection as *the normal vector line* in my discussion below. The normal vector is perpendicular to the cutting plane that divides the Yeas and Nays. Panel B of Figure 3.8 shows the projection of the legislators onto the normal vector line. Technically, these points are on a line through an

Figure 3.8A: Twelve Legislator Example
Normal Vector and Projection Line

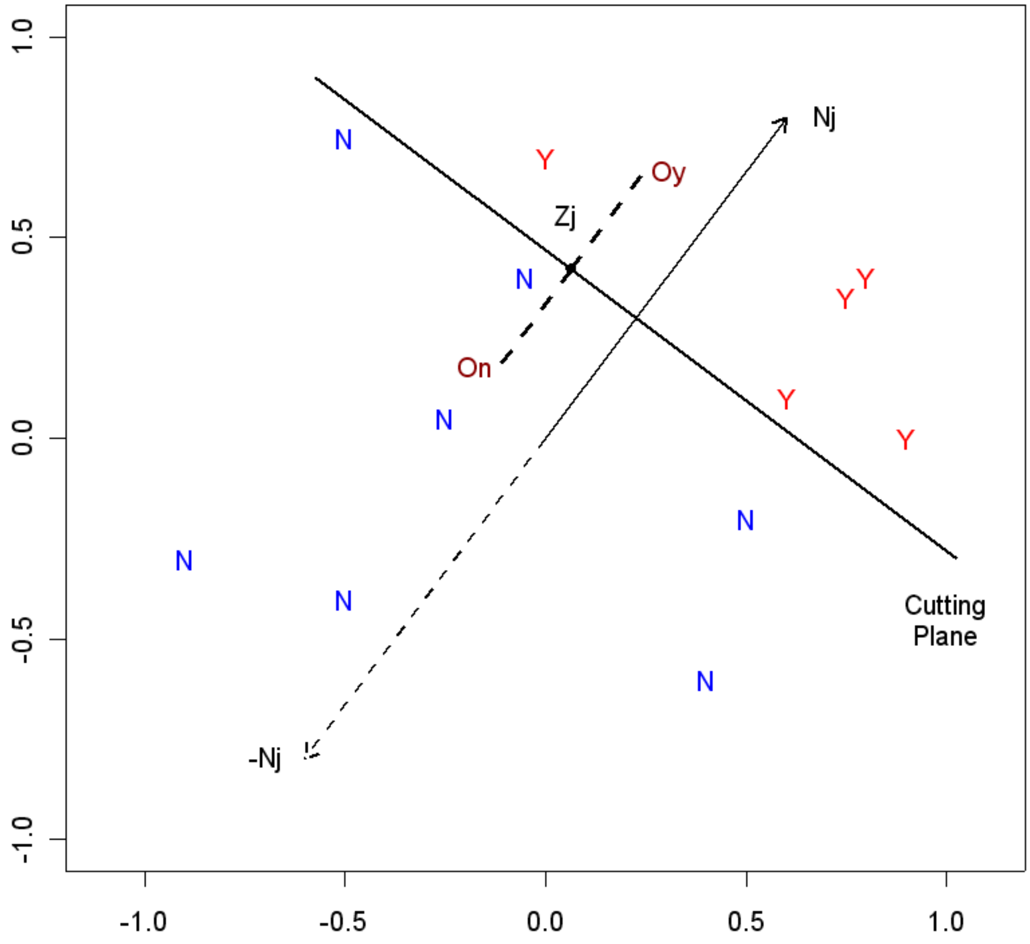
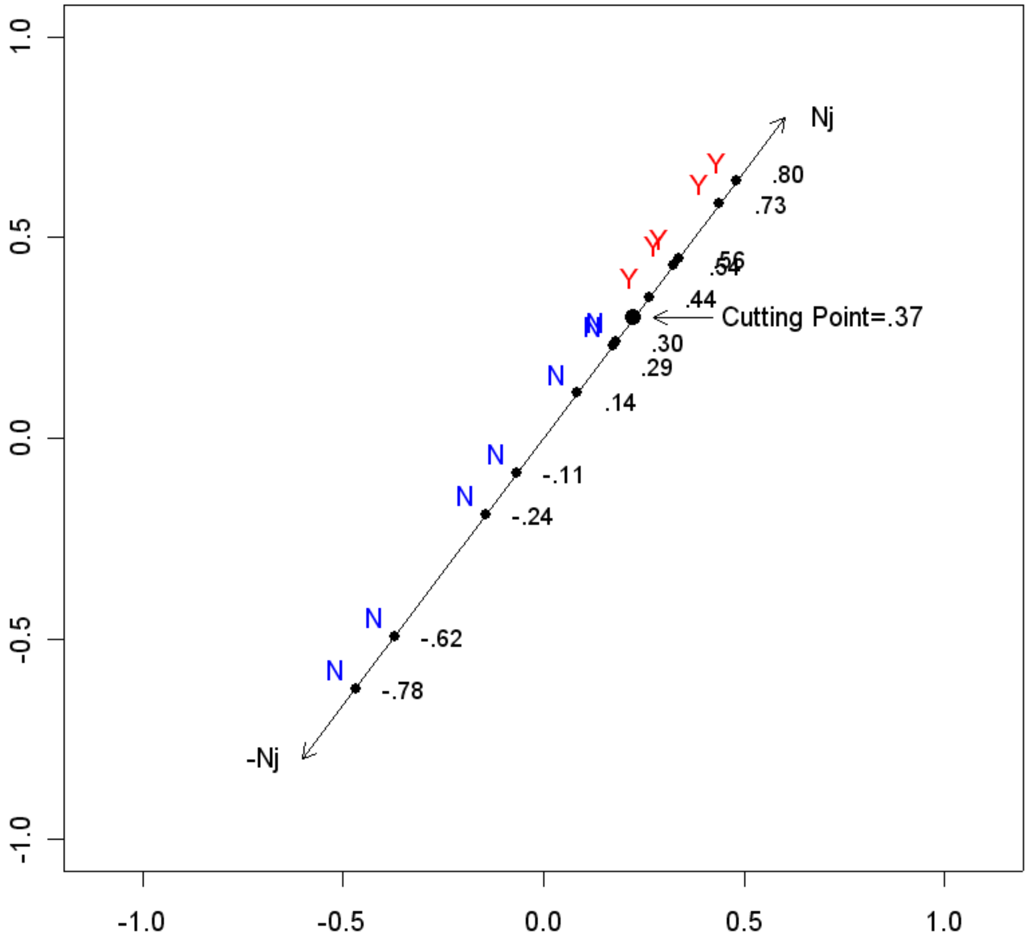


**Figure 3.8B: Twelve Legislator Example**
**Points Projected Onto Projection Line**



s-dimensional space. However, because they *are on a line*, they can be treated *as a line*. The algebra of this is shown in Figure 3.9. I will refer to this as *the projection line* in my discussion below (these values are also shown in Figure 3.8B).

The upper portion of Figure 3.9 shows the legislator coordinates used in Figure 3.8A and those points projected onto the normal vector line. Note that a legislator's point on the normal vector line can be found by simply drawing a line parallel to the cutting plane down to the normal vector line. Technically, this transformation for an individual legislator is $\mathbf{X_i}'\mathbf{N_j}\mathbf{N_j}'$. Note that this produces a vector of length s; that is, a point in s dimensions.

The lower portion of Figure 3.9 shows the legislator positions on the projection line that is just the normal vector line rigidly rotated to the horizontal so that each legislator is simply a single number along the line. Technically, the legislator values along the projection line are:

$$\mathbf{X_i}'\mathbf{N_j} = w_i \qquad (3.3)$$

Where $w_i$ is a scalar – that is, a simple number -- and not a vector. The spacing of the legislators on the projection line *is exactly the same* as their spacing on the normal vector line. The reason for this is straightforward. A legislator's point on the normal vector line is $\mathbf{X_i}'\mathbf{N_j}\mathbf{N_j}'$ or, substituting equation (3.3), $w_i\mathbf{N_j}'$. For two legislators, a and b, the squared distance between them on the projection line is $(w_a - w_b)^2$. The squared distance between them on the normal vector line is:

$$\sum_{k=1}^{s}(w_a N_{jk} - w_b N_{jk})^2 = \sum_{k=1}^{s} N_{jk}^2 (w_a - w_b)^2 = (w_a - w_b)^2 \sum_{k=1}^{s} N_{jk}^2 = (w_a - w_b)^2$$

## Figure 3.9

## Legislator Points Projected onto the Normal Vector and the *Projection Line*

**Legislator Points From Figure 3.8**

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ x_{41} & x_{42} \\ x_{51} & x_{52} \\ x_{61} & x_{62} \\ x_{71} & x_{72} \\ x_{81} & x_{82} \\ x_{91} & x_{92} \\ x_{101} & x_{102} \\ x_{111} & x_{112} \\ x_{121} & x_{122} \end{bmatrix} = \begin{bmatrix} -.90 & -.30 \\ -.50 & -.40 \\ .40 & -.60 \\ -.25 & .05 \\ .50 & -.20 \\ -.05 & .40 \\ -.50 & .75 \\ .60 & .10 \\ .90 & .00 \\ .00 & .70 \\ .75 & .35 \\ .80 & .40 \end{bmatrix}$$

**Legislator Points Projected Onto Normal Vector Line**

$$\begin{bmatrix} -.468 & -.624 \\ -.372 & -.496 \\ -.144 & -.192 \\ -.066 & -.088 \\ .084 & .112 \\ .174 & .232 \\ .180 & .240 \\ .264 & .352 \\ .324 & .432 \\ .336 & .448 \\ .438 & .584 \\ .480 & .640 \end{bmatrix}$$

**Legislator Values on *Projection Line*: $X_i'N_j = w_i$**

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \\ w_{10} \\ w_{11} \\ w_{12} \end{bmatrix} = XN_j = \begin{bmatrix} -.90 & -.30 \\ -.50 & -.40 \\ .40 & -.60 \\ -.25 & .05 \\ .50 & -.20 \\ -.05 & .40 \\ -.50 & .75 \\ .60 & .10 \\ .90 & .00 \\ .00 & .70 \\ .75 & .35 \\ .80 & .40 \end{bmatrix} \begin{bmatrix} .6 \\ .8 \end{bmatrix} = \begin{bmatrix} -.78 \\ -.62 \\ -.24 \\ -.11 \\ .14 \\ .29 \\ .30 \\ .44 \\ .54 \\ .56 \\ .73 \\ .80 \end{bmatrix}$$

so that the distances between the legislators on both lines is exactly the same because the normal vector has unit length (equation 3.2). For example, the distance between legislators eleven and twelve in the two projections shown in Figures 3.9 is

$$\sqrt{(.438 - .480)^2 + (.584 - .640)^2} = .07 = |.73 - .80|$$

Figures 3.8 and 3.9 illustrate the fact that the problem of estimating the roll call cutting plane given the legislator ideal points is equivalent to finding a normal vector, $\mathbf{N_j}$ , such that when the legislator points are projected onto the projection line a cutting point can be found that maximizes the correct classification. Hence, the cutting plane problem has two distinct parts. First, given an estimated normal vector, the plane perpendicular to the normal vector that maximizes correct classifications must be found; and second, given an estimated cutting plane, the orientation of the plane in the space must be changed so that a better estimate of the normal vector is found.

The Janice algorithm solves the first problem. To see this, recall from Chapter 2 that a cutting plane in s dimensions is defined by the vector equation

$$\mathbf{N_j}'(\mathbf{Y} - \mathbf{Z_j}) = \mathbf{0} \qquad (3.4)$$

Where $\mathbf{N_j}$ is the s by 1 unit length normal vector, $\mathbf{Z_j}$ is an s by 1 vector that is the midpoint of the Yea and Nay outcome points, $\mathbf{Y}$ is an s by 1 vector that is *any point on the surface of the plane*, and $\mathbf{0}$ is a s by 1 vector of zeroes. In general, if $\mathbf{Y_A}$ and $\mathbf{Y_B}$ are both points in the plane then, $\mathbf{Y_A}'\mathbf{N_j} = \mathbf{Y_B}'\mathbf{N_j} = c_j$, and $c_j$ is a *scalar* not a vector. By definition, the midpoint of the Yea and Nay outcome points projects to $c_j$; namely,

$$\mathbf{Z_j}'\mathbf{N_j} = c_j \qquad (3.5)$$

Given the $w_i$'s from equation (3.3), finding the optimal $c_j$ is equivalent to the one-dimensional optimal classification problem for a single roll call shown above. The Janice algorithm will find the optimal position for $c_j$ given the $w_i$'s using exactly the same logic displayed in Figures 3.5 and 3.7. Geometrically this is equivalent to moving the cutting plane through the space along the normal vector and counting the correct classifications every time the plane passes through a legislator point.

In this application of the Janice algorithm, real numbers rather than *ranks* are used. For example, consider the simple example shown in Figures 3.8 and 3.9. Suppose the cutting point is placed .01 units to the left of legislator One on the projection line – in this case at -.79. With Nay to the right of the cutting point, this predicts 12 Nays and 0 Yeas or five classification errors. With Yea to the right of the cutting point the prediction is 12 Yeas and 0 Nays for seven classification errors. Proceeding as in Figure 3.9, placing the cutting point midway between Legislators One and Two on the projection line, -.70 in this instance, produces six classification errors with Nay to the right of the cutting point and six errors with Yea to the right of the cutting point. Proceeding inward, placing the cutting point between legislators Seven and Eight with Yea to the right of the cutting point produces perfect classification.

To solve the second part of the cutting plane problem – changing the orientation of the plane in the space to get a better estimate of the normal vector -- the cutting plane must be moved through the space in a direction that increases correct classification. This is accomplished by moving the cutting plane towards the legislator points that are classification errors.

To do this, a matrix is created by projecting all the *correctly classified* legislator points onto the surface of the current cutting plane while leaving the incorrectly classified legislators at their original positions. In two dimensions this produces a line through the space made up of correctly classified legislators (the current cutting plane) around which is a scattering of points corresponding to the incorrectly classified legislators (see Figure 3.10). Much in the spirit of the classic ordinary least squares regression problem, a new cutting plane can then be estimated by simply finding the plane that best fits this set of points using the principle of least squares. The normal vector to this new plane is the new normal vector.

Specifically, a p by s matrix, $\boldsymbol{\Psi}$, is constructed as follows: if legislator i is correctly classified, then her point is projected onto the cutting plane and that point becomes the ith row of $\boldsymbol{\Psi}$; if legislator i is incorrectly classified, then her point remains at its original position and that point becomes the ith row of $\boldsymbol{\Psi}$. That is:

$$\begin{aligned} \boldsymbol{\psi}_i &= \mathbf{X}_i + (c_j - w_i)\mathbf{N}_j \quad \text{if correctly classified} \\ \boldsymbol{\psi}_i &= \mathbf{X}_i \quad \text{if incorrectly classified} \end{aligned} \qquad (3.6)$$

where $\boldsymbol{\psi}_i$ is an s by 1 vector that is the ith row of $\boldsymbol{\Psi}$. In the correctly classified case, to see that $\boldsymbol{\psi}_i$ is on the plane, note that just as in equation (3.5):

$$\mathbf{N}_j'\boldsymbol{\psi}_i = \mathbf{N}_j'\mathbf{X}_i + (c_j - w_i)\mathbf{N}_j'\mathbf{N}_j = w_i + (c_j - w_i) = c_j$$

because $\mathbf{N}_j'\mathbf{N}_j = 1$.

Figures 3.10A and 3.10B show the projection of the legislator ideal points from Figure 3.8 onto a normal vector that is perpendicular to the first dimension; namely, $\mathbf{N}_j = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$. There are three legislators that are not on the correct side of the cutting plane – two legislators who actually vote Nay but are predicted to vote Yea and one legislator who actually votes Yea but is predicted to vote Nay. These three are not projected on to the cutting plane.

**Figure 3.10A: Cutting Plane Procedure Projecting Points onto Cutting Line**

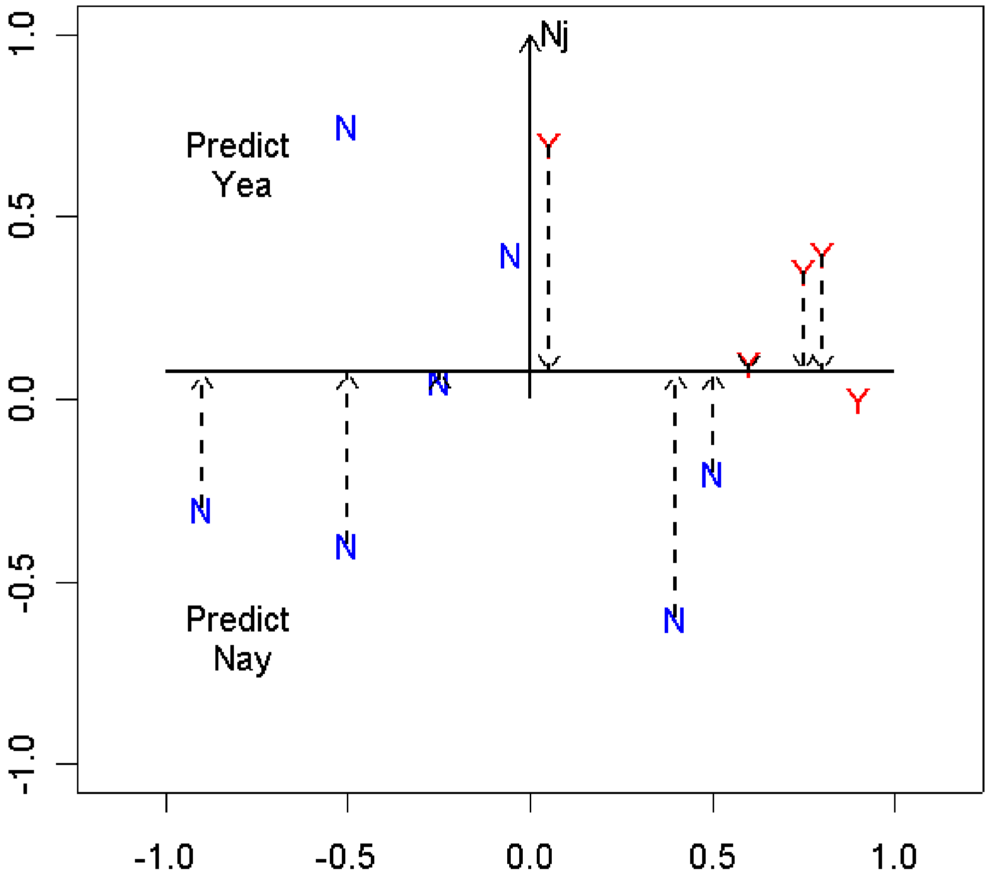
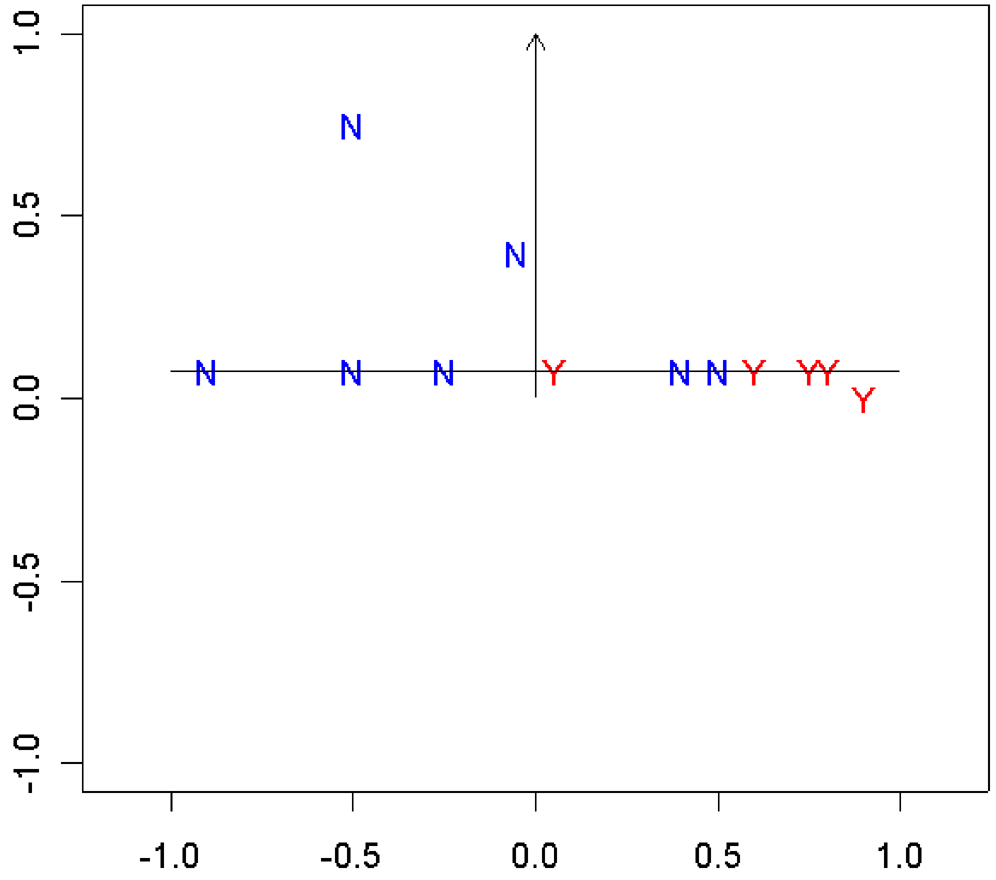


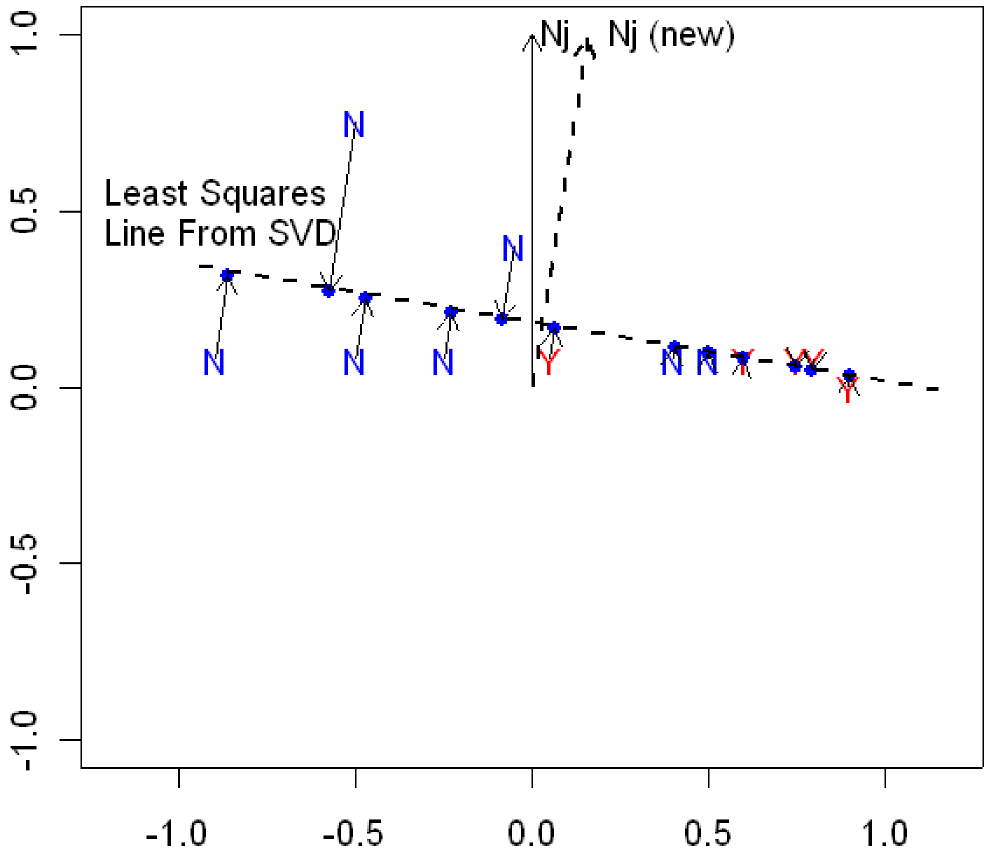
Figure 3.10B: Cutting Plane Procedure
N = (0.000 1.000), 1st Iteration



It is clear from Figures 3.10A and 3.10B that the cutting plane must rotate clockwise towards the errors for the classification error to be reduced. This is achieved by finding the line through the points in Figure 3.10B that minimizes the sum of the squared distances between the points and the line. The new normal vector is perpendicular to this least squares line and the new cutting line is found using the Janice algorithm. The new cutting line is parallel to the least squares line.

Figure 3.10C shows the least squares line through the points from Figure 3.10B. Note that the points are projected *orthogonally* onto the least squares line. This should not be confused with the way a simple linear regression line is represented. In OLS one of the dimensions *is the dependent variable* and the projection to the regression line – quite literally the residual – is *parallel* to the dimension representing the dependent variable. In Figure 3.10C there is no dependent variable. Nevertheless, it is a least squares problem.[3]

**Figure 3.10C: Cutting Plane Procedure**
**Least Squares Line, 1st Iteration**



The normal vector to the least squares line (or plane in more than two dimensions) shown in Figure 3.10C is the new normal vector. The least squares plane is easily estimated. First the means of columns of $\boldsymbol{\Psi}$ are subtracted from the elements of the corresponding columns and then the *singular value decomposition* (SVD) is computed for this mean centered matrix. In a singular value decomposition a matrix of real numbers is written as the product of two orthogonal matrices and one diagonal matrix. (See the Appendix for a discussion of SVD.) The SVD yields the least squares plane and the new normal vector. Figure 3.11 shows how this is done.

Technically, let $\boldsymbol{\mu}$ be the s length vector of the means of the columns of $\boldsymbol{\Psi}$, and let $\mathbf{J_p}$ be a p by 1 vector of ones. Define $\boldsymbol{\Psi}^*$ as

$$\boldsymbol{\Psi}^* = \boldsymbol{\Psi} - J_p\boldsymbol{\mu}' \qquad (3.7)$$

The top portion of Figure 3.11 shows the original legislator points, the points projected on the initial cutting line in Figure 3.10B, and the points with the coordinate means subtracted. The coordinate means are $\mu = \begin{bmatrix} \mathbf{.15} \\ \mathbf{.1521} \end{bmatrix}$. The middle portion of Figure 3.11 shows the singular value decomposition of $\boldsymbol{\Psi}^*$**.** The SVD of $\boldsymbol{\Psi}^*$ is:

$$\boldsymbol{\Psi}^* = \mathbf{U\Lambda V'} \qquad (3.8)$$

where $\mathbf{U}$ is a p by s orthogonal matrix, $\mathbf{V}$ is an s by s orthogonal matrix, and $\boldsymbol{\Lambda}$ is an s by s diagonal matrix containing the *singular values* in descending order on the diagonal. By definition, $\mathbf{U'U = V'V = I_s}$, where $\mathbf{I_s}$ is an s by s identity matrix (see Appendix).

## Figure 3.11

## Cutting Plane Procedure: Projection of Legislator Points onto Cutting Plane

**Legislator Points**

$$
\begin{array}{ccc}
\text{Fig. 3.10A} & \text{Fig. 3.10B} & \text{Means Subtracted} \\
\begin{bmatrix}
-.90 & -.30 \\
-.50 & -.40 \\
.40 & -.60 \\
-.25 & .05 \\
.50 & -.20 \\
-.05 & .40 \\
-.50 & .75 \\
.60 & .10 \\
.90 & .00 \\
.05 & .70 \\
.75 & .35 \\
.80 & .40
\end{bmatrix} &
\begin{bmatrix}
-.90 & .075 \\
-.50 & .075 \\
.40 & .075 \\
-.25 & .075 \\
.50 & .075 \\
-.05 & .400 \\
-.50 & .750 \\
.60 & .075 \\
.90 & .000 \\
.05 & .075 \\
.75 & .075 \\
.80 & .075
\end{bmatrix} &
\begin{bmatrix}
-1.05 & -.0771 \\
-.65 & -.0771 \\
.25 & -.0771 \\
-.40 & -.0771 \\
.35 & -.0771 \\
-.20 & .2479 \\
-.65 & .5979 \\
.45 & -.0771 \\
.75 & -.1521 \\
-.10 & -.0771 \\
.60 & -.0771 \\
.65 & -.0771
\end{bmatrix}
\end{array}
$$

**Singular Value Decomposition of Ψ***

$$
\begin{bmatrix}
-1.05 & -.0771 \\
-.65 & -.0771 \\
.25 & -.0771 \\
-.40 & -.0771 \\
.35 & -.0771 \\
-.20 & .2479 \\
-.65 & .5979 \\
.45 & -.0771 \\
.75 & -.1521 \\
-.10 & -.0771 \\
.60 & -.0771 \\
.65 & -.0771
\end{bmatrix}
= U\Lambda V' =
\begin{bmatrix}
-.5135 & -.3811 \\
-.3156 & -.2815 \\
.1299 & -.0572 \\
-.1918 & -.2192 \\
.1793 & -.0323 \\
-.1187 & .3345 \\
-.3692 & .7650 \\
.2288 & -.0074 \\
.3833 & -.0489 \\
-.0434 & -.1444 \\
.3031 & .0300 \\
.3278 & .0425
\end{bmatrix}
\begin{bmatrix}
1.9950 & .0 \\
.0 & .6368
\end{bmatrix}
\begin{bmatrix}
.9873 & -.1587 \\
.1587 & .9873
\end{bmatrix}
$$

**Figure 3.11 (Cont.)**

**Legislator Values on Least Squares Line (Eckart-Young Theorem) in Fig. 3.10C**

$$\begin{bmatrix} -.8615 & .3146 \\ -.4716 & .2520 \\ .4058 & .1110 \\ -.2279 & .2128 \\ .5033 & .0953 \\ -.0838 & .1897 \\ -.5773 & .2690 \\ .6007 & .0796 \\ .9049 & .0308 \\ .0646 & .1658 \\ .7470 & .0561 \\ .7957 & .0483 \end{bmatrix} = \begin{bmatrix} -.5135 & -.3811 \\ -.3156 & -.2815 \\ .1299 & -.0572 \\ -.1918 & -.2192 \\ .1793 & -.0323 \\ -.1187 & .3345 \\ -.3692 & .7650 \\ .2288 & -.0074 \\ .3833 & -.0489 \\ -.0434 & -.1444 \\ .3031 & .0300 \\ .3278 & .0425 \end{bmatrix} \begin{bmatrix} 1.9950 & .0 \\ .0 & .0 \end{bmatrix} \begin{bmatrix} .9873 & -.1587 \\ .1587 & .9873 \end{bmatrix} + \begin{bmatrix} .15 \\ .1521 \end{bmatrix}$$

The bottom portion of Figure 3.11 shows the least squares solution – the dotted line in Figure 3.10C. The line is found by applying the famous Eckart-Young theorem (Eckart and Young, 1936 – see Appendix for a formal statement of the theorem) to the points placed at their means – that is, to $\boldsymbol{\Psi}^*$. The least squares line is found by inserting a zero in place of the second singular value (the smallest value) in $\boldsymbol{\Lambda}$**,** remultiplying, and then adding the means back to the columns of the reconstructed matrix. That is, let $\Lambda^{\#}$ be the s by s diagonal matrix identical to $\Lambda$ except for the replacement of the sth singular value by zero (that is, the smallest value), then the estimated line (in two dimensions) or plane (in three or more dimensions) is:

$$\mathbf{Y} = \mathbf{U}\boldsymbol{\Lambda}^{\#}\mathbf{V}' + \mathrm{J}_{\mathrm{p}}\boldsymbol{\mu}' \qquad (3.9)$$

Where the p by s matrix **Y** will have rank s-1 by construction. In two dimensions the rows of **Y** form a line as in Figure 3.10C. In three dimensions, the rows of **Y** are points lying on a two-dimensional plane.

The new normal vector is perpendicular to this least squares line (see Figure 3.10C). By definition, this new normal vector is the sth singular vector (sth column) in **V** corresponding to the smallest singular value. In this case the new normal vector is $\mathbf{N}_j = \begin{bmatrix} \mathbf{.1587} \\ \mathbf{.9873} \end{bmatrix}$ (recall that **V** is shown as transposed in Figure 3.11) so that for every point (row) in **Y**; $\mathbf{Y_i}'\mathrm{N_j} = \mathrm{k}$, where $\mathrm{k} = \boldsymbol{\mu}'\mathbf{N_j}$. For example, for the first row of **Y** in Figure 3.11 this product is:

$\mathbf{Y_1}'\mathrm{N_j} = -.8615*.1587 + .3146*.9873 = .1739$ and $\boldsymbol{\mu}'\mathbf{N_j} = .15*.1587 + .1521*.9873 = .1739$ (there is some rounding error).

Figure 3.12A shows the new normal vector and cutting line. As explained above, the new cutting line is found using the Janice algorithm and it is parallel to the least squares line in Figure 3.10C. Figure 3.12A shows that the cutting plane must again be rotated clockwise towards the two errors for the classification error to be reduced. Once again this is achieved by finding the line through the points in Figure 3.12A that minimizes the sum of the squared distances between the points and the line. The new normal vector is perpendicular to this least squares line and the new cutting line is found using the Janice algorithm. This process can be continued until there is no further improvement in classification error. Figure 3.12B shows the process at the 5th iteration and Figure 3.12C shows the process at the 7th iteration when perfect classification is achieved.

Figure 3.12A: Cutting Plane Procedure
N = (0.157  0.988), 2nd Iteration

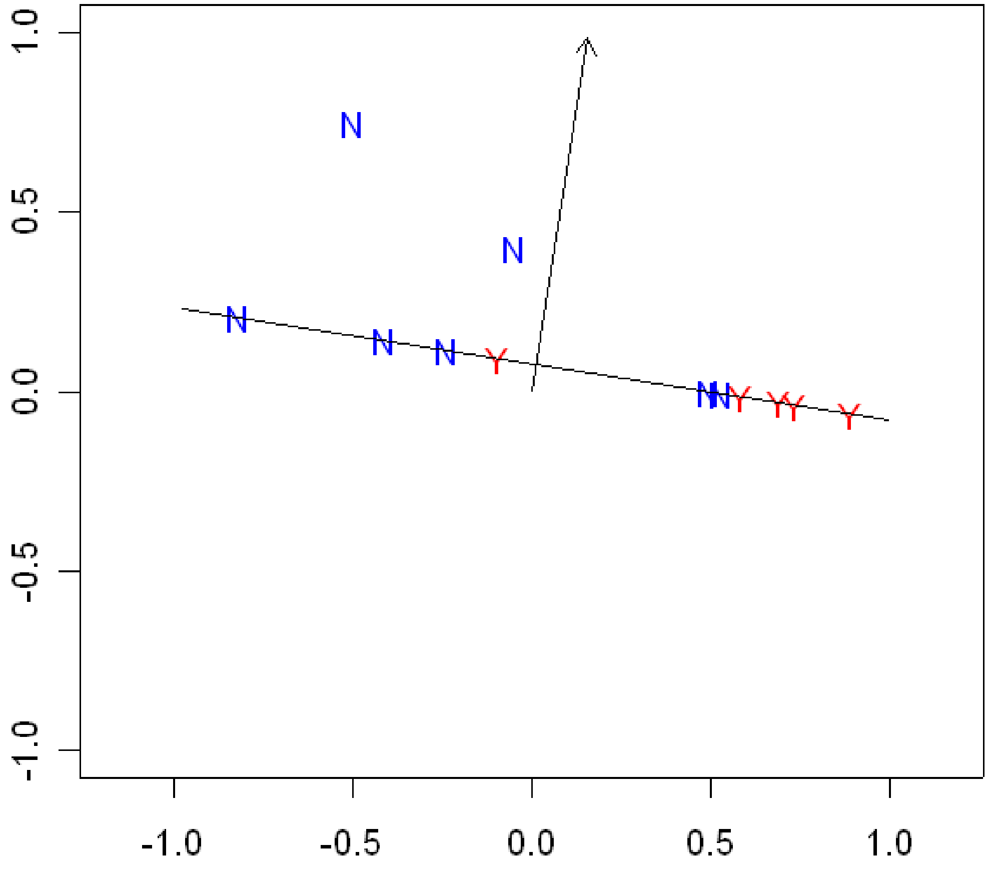


Figure 3.12B: Cutting Plane Procedure
N = (0.492  0.871), 5th Iteration

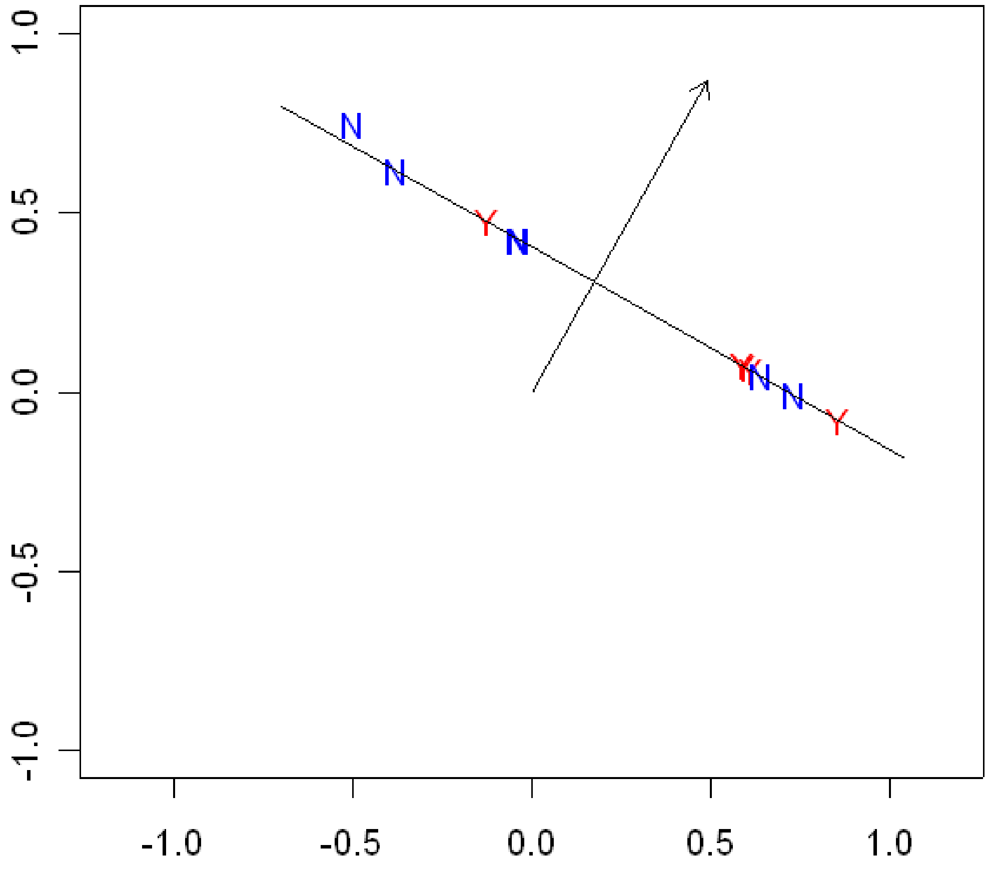
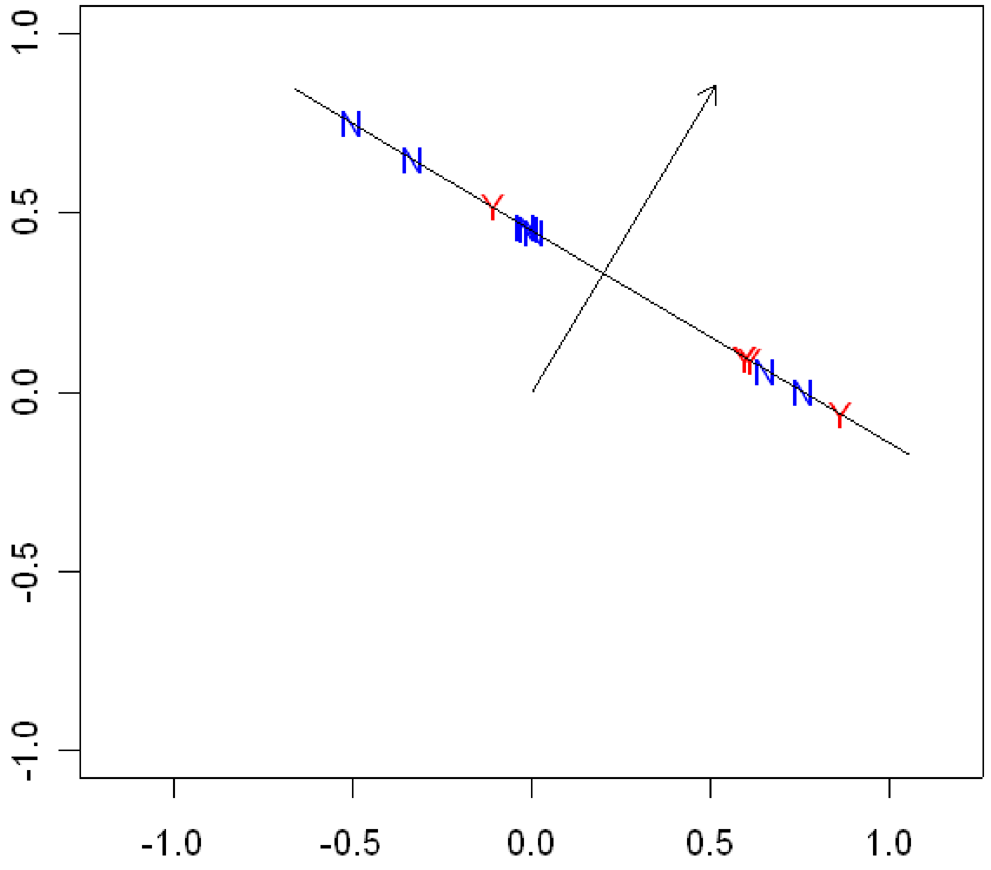


Figure 3.12C: Cutting Plane Procedure
N = (0.514  0.858), 7th Iteration



In sum, calculating the optimal $\mathbf{N_j}$ consists of the following steps:

1) Obtain a starting estimate of $\mathbf{N_j}$ (more on the practical aspects of this in Chapter 5).
2) Calculate the correct classifications associated with $\mathbf{N_j}$.
3) Construct $\boldsymbol{\Psi}^*$ using equations (3.6) and (3.7).
4) Perform singular value decomposition of $\boldsymbol{\Psi}^*$, $\mathbf{U\Lambda V'}$ .

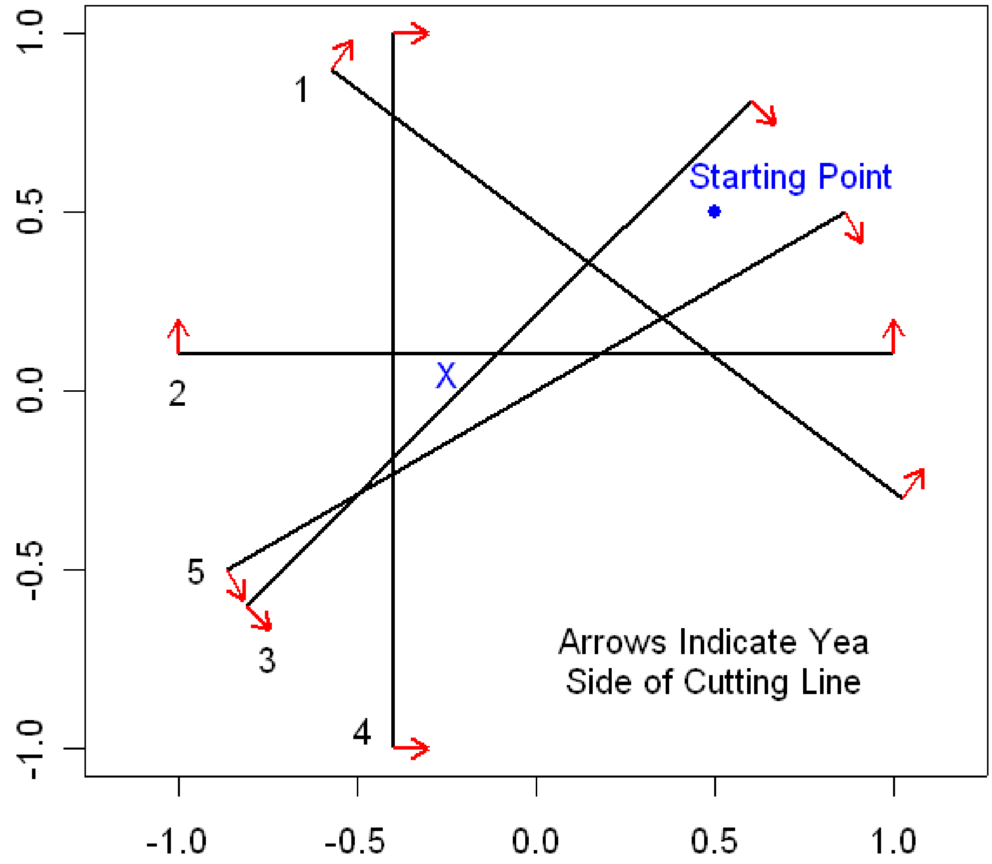
5) Use the sth singular vector (sth column) of **V**, $\mathbf{v}_s$ , as the new estimate of $\mathbf{N_j}$.
6) Go to (2).

In a perfect case like that shown in Figure 3.8, the cutting plane procedure will almost always quickly iterate into the true cutting plane. In Poole (2000) I report an extensive set of Monte Carlo experiments showing that the cutting plane procedure performs very well.[4] In particular, when error is present the cutting plane procedure is highly accurate and converges very quickly. When there is error the converged cutting plane may not be the one that maximizes correct classification because the cutting plane procedure is minimizing the sum of squared distances of the classification errors to the cutting plane. However, the converged cutting plane is almost always very close to the optimal cutting plane. This is easily dealt with by simply storing the iteration record of the cutting plane procedure and using the normal vector corresponding to the best classification. I have found that this works very well in practice. The procedure does a good job correctly classifying the true roll call choices and recovering the true normal vectors – especially at the 15 percent error level which is the approximate level of the error found in the U.S. Congressional roll call data.[5] Finally, as one would expect, increasing the number of legislators increases the accuracy of the recovery.

## Estimating the Legislator Ideal Points Given the Roll Call Cutting Planes

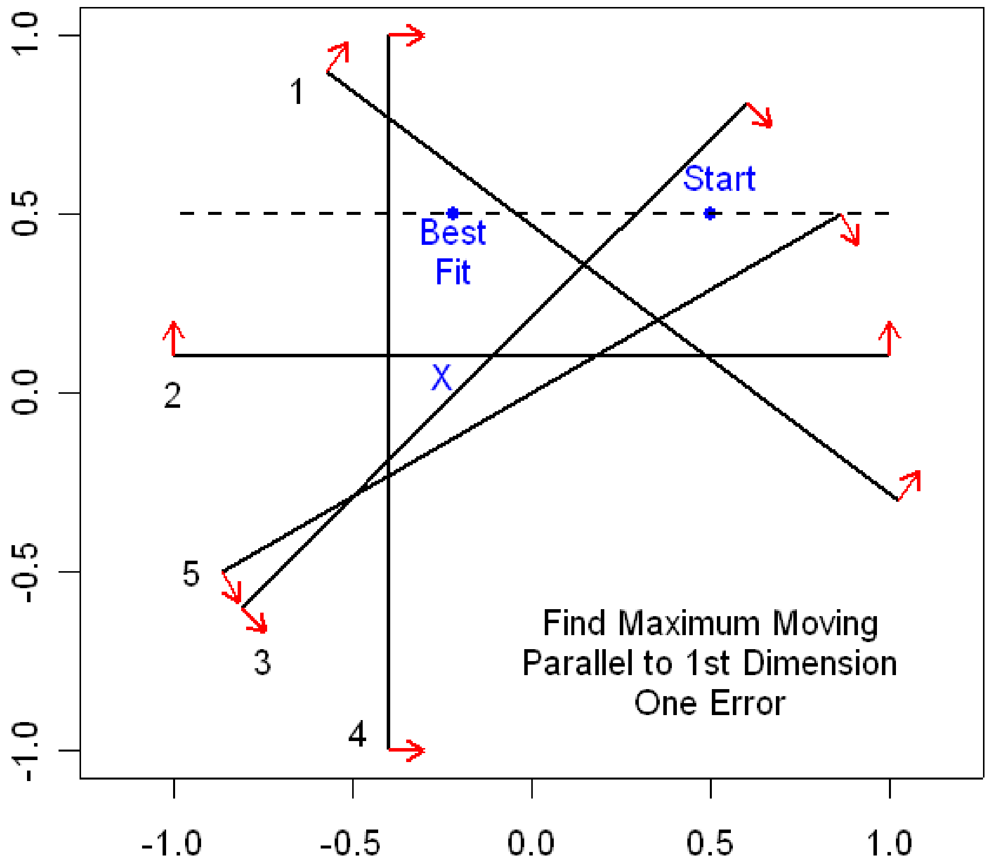
In Chapter 2 I show that in the geometry of the roll call voting problem *legislators are not points they are polytopes*. Given the q cutting lines/planes corresponding to the q normal vectors and the votes of the ith legislator on the q roll calls, the problem is to find the polytope that maximizes the correct classification. The legislator's ideal point, $\mathbf{X_i}$, is then placed within this polytope. Figure 3.13 shows an example in two dimensions.

Figure 3.13A: Locating the Legislator NNNYN Polytope



Arrows Indicate Yea Side of Cutting Line

Figure 3.13B: Locating the Legislator NNNYN Polytope



Find Maximum Moving Parallel to 1st Dimension One Error

Figure 3.13C: Locating the Legislator NNNYN Polytope

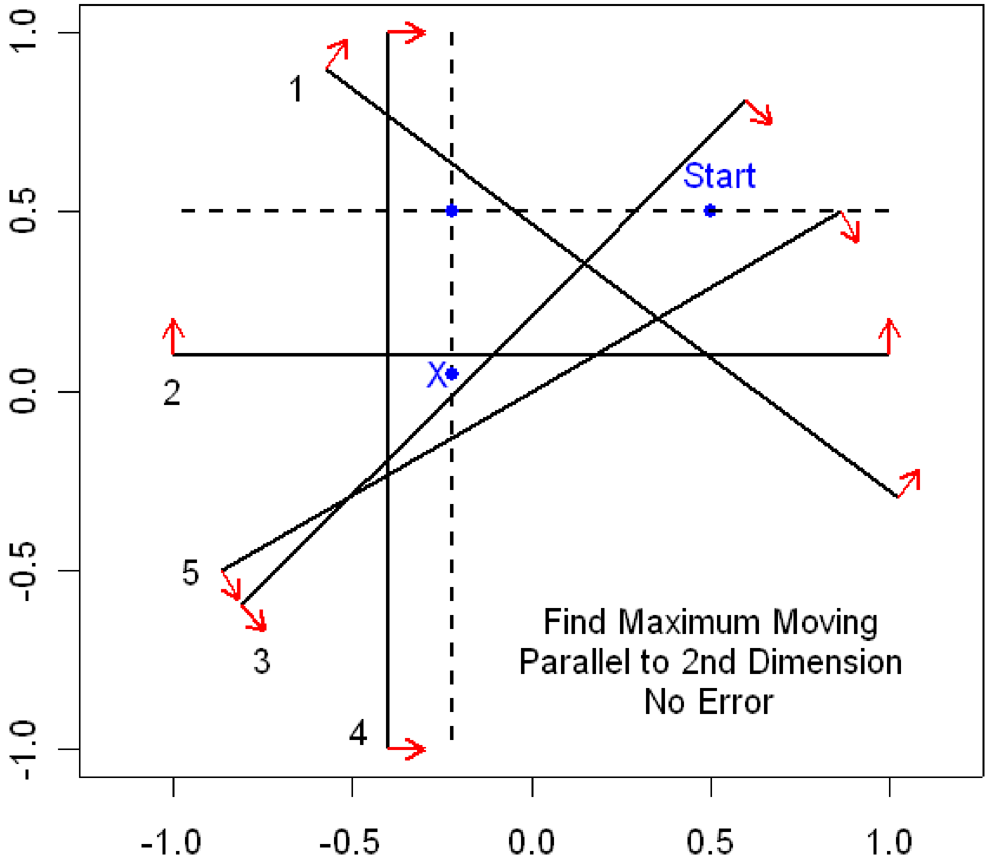


Figure 3.13 shows five cutting lines indicated by the numbering at the ends of the cutting lines. The arrows at the ends of the cutting lines indicate the Yea side of the cutting line. The five cutting lines produce a complete Coombs Mesh because the number of regions in Figure 3.13 is 16 ($\sum_{k=0}^{s}\binom{q}{k} = \binom{5}{0}+\binom{5}{1}+\binom{5}{2}=16$).

Given a legislator's pattern of votes, in this case NNNYN, the problem is to find the polytope in Figure 3.13 that maximizes the correct classification. In this example the point "X" is located in the NNNYN polytope corresponding to perfect classification.

Suppose the initial estimate of the legislator's coordinates is at the point labeled "starting point" in Figure 3.13A. This initial estimate is in the open polytope YYYYN corresponding to only two correct classifications (roll calls 4 and 5). The problem is to move the point representing the legislator in a direction that increases the number of correct classifications.

Below a method is shown for finding the maximum classification point along any arbitrary line passing through the space. This method is used to move the legislator point through the space in a city-block fashion by searching along a line parallel to the first dimension for the polytope that maximizes correct classification. The legislator point is then placed inside the best polytope on the search line. For example, in Figure 3.13B the dotted line through the starting point (from right to left) passes through the polytopes: NYNNN, NYNYN, YYNYN, YYYYN (starting polytope), and YYYYY (the open polytope adjacent to the starting polytope). The NYNYN is the best fit with only one classification error – the legislator is on the wrong side of the cutting line for roll call number 2. The "best fit" point is placed on the search line midway through the polytope.

The next step is to move the legislator point along a line through this new point but parallel to the second dimension. Again the best polytope is found and the legislator point is placed inside this polytope on the search line. For example, in Figure 3.13C the dotted line through the starting point (from top to bottom) passes through the polytopes: YYNYN, NYNYN (current position of legislator point), NNNYN, NNYYN, and NNYYY. The NNNYN is a perfect fit with no classification error and the legislator is placed at the point "X".

This process can be repeated as many times as desired. For example, suppose that NNNYN was simply the best fit along the vertical dotted line but there was still incorrect classifications. A line parallel to the first dimension passing through "X" would now be searched. This line would pass through the polytopes NNNNN, NNNYN, NNYYN, NNYYY, and YNYYY.

This city-block-like search procedure *always* converges to a point inside a polytope for which the coordinates are at a *local maximum* in terms of classification. That is, the point cannot be moved parallel to any dimension and have the correct classifications increase.

The search procedure is constructed as follows. Let $\mathbf{X_i}^{(h)}$ be the initial estimate for legislator i where "h" is the iteration number (1, 2, 3, etc.) and let $\mathbf{X_i}^{(a)}$ be a second point. The problem is to find a new estimate, $\mathbf{X_i}^{(h+1)}$, on the line passing through $\mathbf{X_i}^{(h)}$ and $\mathbf{X_i}^{(a)}$ that is inside a polytope that increases correct classification as illustrated in Figure 3.13. This is done by analyzing the projections of $\mathbf{X_i}^{(h)}$ and $\mathbf{X_i}^{(a)}$ onto the q normal vectors.

The projection of $\mathbf{X_i}^{(h)}$ onto the jth normal vector is:

$$\mathbf{X_i}^{(h)\prime}\mathbf{N_j} = w_{ij}^{(h)} \tag{3.10}$$

Similarly, the projection of the second point onto the jth normal vector is $w_{ij}^{(a)}$. These projections correspond to a correct classification on roll call j depending upon which side of the cutting point, $c_j$, they fall. There are six possible orderings of $w_{ij}^{(h)}$, $w_{ij}^{(a)}$, and $c_j$. For each ordering there are two possible classification outcomes for a total of 12 cases. Table 3.1 shows each case. The "R" in Table 3.1 is the maximum limit of the dimensions – this is usually set equal to one so that in two dimensions, the space would be the unit circle.

## Table 3.1

| Case | Ordering | Classification | | Limits of $\alpha$ That Correctly Project $x_i^{(h+1)}$ |
|---|---|---|---|---|
| | | h | a | |
| 1. | $-R < c_j < w_{ij}^{(h)} < w_{ij}^{(a)} < +R$ | C[1] | C | $\frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 2. | $-R < c_j < w_{ij}^{(h)} < w_{ij}^{(a)} < +R$ | I | I | $\frac{-R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 3. | $-R < c_j < w_{ij}^{(a)} < w_{ij}^{(h)} < +R$ | C | C | $\frac{R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 4. | $-R < c_j < w_{ij}^{(a)} < w_{ij}^{(h)} < +R$ | I | I | $\frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{-R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 5. | $-R < w_{ij}^{(h)} < w_{ij}^{(a)} < c_j < +R$ | C | C | $\frac{-R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 6. | $-R < w_{ij}^{(h)} < w_{ij}^{(a)} < c_j < +R$ | I | I | $\frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 7. | $-R < w_{ij}^{(a)} < w_{ij}^{(h)} < c_j < +R$ | C | C | $\frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{-R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 8. | $-R < w_{ij}^{(a)} < w_{ij}^{(h)} < c_j < +R$ | I | I | $\frac{R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 9. | $-R < w_{ij}^{(h)} < c_j < w_{ij}^{(a)} < +R$ | C | I | $\frac{-R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 10. | $-R < w_{ij}^{(h)} < c_j < w_{ij}^{(a)} < +R$ | I | C | $\frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 11. | $-R < w_{ij}^{(a)} < c_j < w_{ij}^{(h)} < +R$ | C | I | $\frac{R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |
| 12. | $-R < w_{ij}^{(a)} < c_j < w_{ij}^{(h)} < +R$ | I | C | $\frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j < \frac{-R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$ |

---

[1] “C” is correctly classified; “I” is incorrectly classified.

In case 1 both $\mathbf{X_i}^{(h)}$ and $\mathbf{X_i}^{(a)}$ project to the right of $c_j$ and are on the correct side of the cutting plane for the jth roll call and are therefore correctly classified as indicated by the two classification columns at the center of the Table. Case 2 is the same geometrically only now $\mathbf{X_i}^{(h)}$ and $\mathbf{X_i}^{(a)}$ are on the wrong side of the cutting plane and are therefore projected as classification errors. Cases 1 to 8 represent no change in classification from moving the legislator point along the line from $\mathbf{X_i}^{(h)}$ to $\mathbf{X_i}^{(a)}$. For $\mathbf{X_i}^{(a)}$ to be an improvement over $\mathbf{X_i}^{(h)}$ , the number of cases 10 and 12 must be greater than the number of cases 9 and 11.

Consider the effect of moving $\mathbf{X_i}^{(h)}$ away from $\mathbf{X_i}^{(a)}$ . This has no effect on cases 3 - 6, and 9 - 12. Only those cases where $\mathbf{X_i}^{(h)}$ is between $\mathbf{X_i}^{(a)}$ and $c_j$ – cases 1, 2, 7, and 8 – are affected. Depending upon how far $\mathbf{X_i}^{(h)}$ is moved away from $\mathbf{X_i}^{(a)}$ , case 1 could change to case 10 increasing the error by one, case 2 could change to case 9 decreasing the error by one, case 7 could change to case 12 increasing the error by one, and case 8 could change to case 11 decreasing the error by one. A similar analysis of the effect of moving $\mathbf{X_i}^{(h)}$ towards $\mathbf{X_i}^{(a)}$ can also be done.

More generally, consider the line equation:

$$\mathbf{X_i}^{(h+1)} = \mathbf{X_i}^{(h)} + \alpha(\mathbf{X_i}^{(a)} - \mathbf{X_i}^{(h)}) \qquad (3.11)$$

which, when projected onto the jth normal vector, becomes:

$$w_{ij}^{(h+1)} = w_{ij}^{(h)} + \alpha(w_{ij}^{(a)} - w_{ij}^{(h)}) \qquad (3.12)$$

If $\alpha=0$ then $\mathbf{X_i}^{(h)} = \mathbf{X_i}^{(h+1)}$ and the legislator point is not moved. If $\alpha>0$ then $\mathbf{X_i}^{(h)}$ is moved towards $\mathbf{X_i}^{(a)}$ and if $\alpha<0$ then $\mathbf{X_i}^{(h)}$ is moved away from $\mathbf{X_i}^{(a)}$. For a single roll call, it is easy to solve for $\alpha$; these are shown in Table 2 for all 12 cases. For example, for case 2,

$\alpha$ must be chosen so that the projection of $\mathbf{X_i}^{(h+1)}$, $w_{ij}^{(h+1)}$, is in the region $(-R, c_j)$. Solving for $\alpha$:

$$-R = w_{ij}^{(h)} + \alpha(w_{ij}^{(a)} - w_{ij}^{(h)}) \text{ so that } \frac{-R - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}} < \alpha_j$$

$$c_j = w_{ij}^{(h)} + \alpha(w_{ij}^{(a)} - w_{ij}^{(h)}) \text{ so that } \alpha_j < \frac{c_j - w_{ij}^{(h)}}{w_{ij}^{(a)} - w_{ij}^{(h)}}$$

Given $\mathbf{X_i}^{(h)}$ and $\mathbf{X_i}^{(a)}$, Table 3.1 can be used to find the limits of $\alpha$ for each roll call. Let the upper and lower limits for the jth roll call be $U_{ij}$ and $L_{ij}$ respectively. The correct classification associated with $\mathbf{X_i}^{(h)}$ can be obtained by setting $\alpha=0$ and counting the number of roll calls for which $0 \in (L_{ij}, U_{ij})$. Similarly, the correct classification associated with $\mathbf{X_i}^{(a)}$ is obtained by setting $\alpha=1$ and counting the number of roll calls for which $1 \in (L_{ij}, U_{ij})$. In general, define

$$\delta_{ij} = 1 \text{ if } \alpha \in (L_{ij}, U_{ij})$$

$$\delta_{ij} = 0 \text{ if } \alpha \notin (L_{ij}, U_{ij})$$

and the correct classification is simply

$$\delta(\alpha) = \sum_{j=1}^{q} \delta_{ij} \tag{3.13}$$

The $\alpha$ that maximizes $\delta(\alpha)$, the number of correct classifications, can be calculated in a simple manner. First, compute the $L_{ij}$ and $U_{ij}$ for each roll call. Second, rank order the $L_{ij}$ and $U_{ij}$ and use the Janice algorithm to calculate the optimal $\alpha$. Here the $L_{ij}$ play the role of "Yea" and the $U_{ij}$ play the role of "Nay". For example, if there exists an $\alpha$ that results in perfect classification, the ordering of L's and U's will look like (dropping the i subscript to reduce clutter and numbering left to right for convenience):

$$L_1 < L_2 < L_3 < \ldots < L_q < U_1 < U_2 < U_3 < \ldots < U_q$$

that is, all the $L_j$ will be less than all the $U_j$ so that with $\alpha \in (L_q , U_1)$ there is perfect classification, $\delta(\alpha) = q$.

For example, using the configuration shown in Figure 3.13, the starting point (h=1) $\mathbf{X_i}^{(1)}$ is placed at .5, .5; that is, $\begin{bmatrix} \mathbf{x}_{11}^{(1)} \\ \mathbf{x}_{12}^{(1)} \end{bmatrix} = \begin{bmatrix} \mathbf{.5} \\ \mathbf{.5} \end{bmatrix}$, and the second point $\mathbf{X_i}^{(a)}$ is placed on the dotted line; specifically, $\begin{bmatrix} \mathbf{x}_{11}^{(a)} \\ \mathbf{x}_{12}^{(a)} \end{bmatrix} = \begin{bmatrix} \mathbf{.01} \\ \mathbf{.5} \end{bmatrix}$. The resulting rank order of the upper and lower limits is:

$$L_4 < L_5 < L_3 < L_1 < U_4 < U_3 < U_5 < U_1$$

Numerically:

$$-1.020 < -.747 < .433 < 1.105 < 1.837 < 2.887 < 3.335 < 5.782$$

For the first roll call $w_{11}^{(h)} = \begin{bmatrix} \mathbf{.5} & \mathbf{.5} \end{bmatrix} \begin{bmatrix} \mathbf{.6} \\ \mathbf{.8} \end{bmatrix} = .7$, $w_{11}^{(a)} = \begin{bmatrix} \mathbf{.01} & \mathbf{.5} \end{bmatrix} \begin{bmatrix} \mathbf{.6} \\ \mathbf{.8} \end{bmatrix} = .406$, and $c_j = .375$ so that $c_j < w_{11}^{(a)} < w_{11}^{(h)}$. The hypothetical legislator votes Nay on the first roll call. However, both $\mathbf{X_i}^{(h)}$ and $\mathbf{X_i}^{(a)}$ are on the Yea side of the cutting line (see Figure 3.13B) so that $w_{11}^{(h)}$ and $w_{11}^{(a)}$ are both incorrect classifications in Table 3.1 This is an example of Case 4. Using R=1, the upper and lower limits are $\frac{-1-.7}{.406-.7} = 5.782$ and $\frac{.375-.7}{.406-.7} = 1.105$.

Note that $L_2$ and $U_2$ do not appear in the ordering. This occurs because the normal vector for the second roll call is $\begin{bmatrix} \mathbf{.0} \\ \mathbf{1.0} \end{bmatrix}$ so that $w_{i2}^{(h)} = w_{i2}^{(a)}$. In other words, if the line through $\mathbf{X_i}^{(h)}$ and $\mathbf{X_i}^{(a)}$ is parallel to a cutting line then $w_{ij}^{(a)} - w_{ij}^{(h)}$, which is used in

Table 3.1 to find $\alpha_j$, is equal to zero. This is not a problem because when the line through $\mathbf{X_i}^{(h)}$ and $\mathbf{X_i}^{(a)}$ is parallel to a cutting line then the classification on the corresponding roll call is the same no matter where on the line $\mathbf{X_i}^{(h+1)}$ is located.

The rank ordering above is a perfect pattern but there is still one classification error because both $\mathbf{X_i}^{(h)}$ and $\mathbf{X_i}^{(a)}$ are on the Yea side of the cutting line for the second roll call and it was *omitted* from the rank ordering. Consequently, the point resulting from using $\alpha \in (L_1 = 1.105, U_4 = 1.837)$ , $\mathbf{X_i}^{(2)}$ , the "Best Fit" point in Figure 3.13B, has one classification error with 4 correct classifications. (In practice, $\alpha$ is set equal to the midpoint; in this case, $(L_1 + U_4)/2 = 1.4711$ [there is rounding error].) Using equation (3.11), the "Best Fit" point is: $\begin{bmatrix} \mathbf{.5} \\ \mathbf{.5} \end{bmatrix} + \mathbf{1.4711}\left(\begin{bmatrix} \mathbf{.01} \\ \mathbf{.5} \end{bmatrix} - \begin{bmatrix} \mathbf{.5} \\ \mathbf{.5} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{-.221} \\ \mathbf{.5} \end{bmatrix}$.

For the second iteration, h=2, the starting estimate is $\begin{bmatrix} \mathbf{x_{11}^{(2)}} \\ \mathbf{x_{12}^{(2)}} \end{bmatrix} = \begin{bmatrix} \mathbf{-.221} \\ \mathbf{.5} \end{bmatrix}$ and the second point is $\begin{bmatrix} \mathbf{x_{11}^{(a)}} \\ \mathbf{x_{12}^{(a)}} \end{bmatrix} = \begin{bmatrix} \mathbf{-.221} \\ \mathbf{.01} \end{bmatrix}$. This produces the rank ordering:

$$L_3 < L_5 < L_1 < L_2 < U_3 < U_5 < U_2 < U_1$$

$L_4$ and $U_4$ are missing because the classification line is parallel to the cutting line for roll call number 4 (see Figure 3.13C). The rank ordering is again a perfect pattern with $\alpha \in (L_2 , U_3 )$ and there are no classification errors because $\mathbf{X_i}^{(2)}$ and $\mathbf{X_i}^{(a)}$ are both on the correct side of the cutting line for roll call number four.

Numerically the value for $\alpha$ is .9272. Using equation 3.11, the solution is $\begin{bmatrix} \mathbf{-.221} \\ \mathbf{.5} \end{bmatrix} + \mathbf{.9272}\left(\begin{bmatrix} \mathbf{-.221} \\ \mathbf{.01} \end{bmatrix} - \begin{bmatrix} \mathbf{-.221} \\ \mathbf{.5} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{-.221} \\ \mathbf{.046} \end{bmatrix}$ which is point "X" in Figure 3.13C.

To recap, the search for the $\mathbf{X_i}$ that maximizes correct classification is conducted in a city-block manner. In the first iteration, the search is along a line through the starting point $\mathbf{X_i}^{(1)}$ and the second point $\mathbf{X_i}^{(a)}$ with all but the first dimension coordinates in $\mathbf{X_i}^{(1)}$ and $\mathbf{X_i}^{(a)}$ set equal to each other so that they lie on a line parallel to the first dimension. In the second iteration, the first dimension coordinates are all set equal to the value corresponding to the optimal first dimension value and the 3rd, 4th, …, $s^{th}$ dimensional coordinates in $\mathbf{X_i}^{(2)}$ and $\mathbf{X_i}^{(a)}$ are set equal to one another so that the points lie on a line parallel to the second dimension. In the third iteration, the first and second dimension coordinates are set equal to the optimal values from the first and second iterations respectively, and the 4th, 5th, …, $s^{th}$ dimensional coordinates in $\mathbf{X_i}^{(3)}$ and $\mathbf{X_i}^{(a)}$ are set equal to one another so that the points lie on a line parallel to the third dimension. This process continues in the same fashion through the $s^{th}$ dimension. Since the search for the optimal $\mathbf{X_i}$ is being done city-block-wise, dimensions 1 to s can now be searched again.

In sum, calculating the optimal $\mathbf{X_i}$ consists of the following steps:

Obtain a realistic starting estimate, $\mathbf{X_i}^{(1)}$ (see Chapter 5).

1) Set $\mathbf{X_i}^{(a)\prime} = [0.01, X_{i2}^{(1)}, X_{i3}^{(1)}, X_{i4}^{(1)}, X_{i5}^{(1)}, \ldots, X_{is}^{(1)}]$, find optimal $\alpha$ and $\mathbf{X_i}^{(2)} = \mathbf{X_i}^{(1)} + \alpha(\mathbf{X_i}^{(a)} - \mathbf{X_i}^{(1)})$ .
2) Set $\mathbf{X_i}^{(a)\prime} = [X_{i1}^{(2)}, 0.01, X_{i3}^{(1)}, X_{i4}^{(1)}, X_{i5}^{(1)}, \ldots, X_{is}^{(1)}]$, find optimal $\alpha$ and $\mathbf{X_i}^{(3)} = \mathbf{X_i}^{(2)} + \alpha(\mathbf{X_i}^{(a)} - \mathbf{X_i}^{(2)})$.
3) Set $\mathbf{X_i}^{(a)\prime} = [X_{i1}^{(2)}, X_{i2}^{(3)}, 0.01, X_{i4}^{(1)}, X_{i5}^{(1)}, \ldots, X_{is}^{(1)}]$, find optimal $\alpha$ and $\mathbf{X_i}^{(4)} = \mathbf{X_i}^{(3)} + \alpha(\mathbf{X_i}^{(a)} - \mathbf{X_i}^{(3)})$.
4) Set $\mathbf{X_i}^{(a)\prime} = [X_{i1}^{(2)}, X_{i2}^{(3)}, X_{i3}^{(4)}, 0.01, X_{i5}^{(1)}, \ldots, X_{is}^{(1)}]$, find optimal $\alpha$ and $\mathbf{X_i}^{(5)} = \mathbf{X_i}^{(4)} + \alpha(\mathbf{X_i}^{(a)} - \mathbf{X_i}^{(4)})$.

etc.

s+1) Set $\mathbf{X_i}^{(a)\prime} = [X_{i1}^{(2)}, X_{i2}^{(3)}, X_{i3}^{(4)}, X_{i4}^{(5)}, \ldots, X_{is-1}^{(s)}, 0.01]$ , find optimal $\alpha$ and $\mathbf{X_i}^{(s+1)} = \mathbf{X_i}^{(s)} + \alpha(\mathbf{X_i}^{(a)} - \mathbf{X_i}^{(s)})$.

s+2) Go to (2).

Note that classification error *can never increase from one step to the next*. This is true because setting $\alpha = 0$ preserves the current value of classification. This process converges very quickly (usually less than 10 iterations through steps 2 to s+1 above) to a vector of coordinates which is a local maximum in terms of classification. That is, it converges to a point such that $\alpha = 0$ for all s dimensions.

In practice, the starting estimate, $\mathbf{X_i}^{(h)}$ , and the second point, $\mathbf{X_i}^{(a)}$ , could be placed anywhere within the s dimensional space. The search does not have to be parallel to any dimension – it can be done along *any line through the space*. Although the search process does not have to be done by moving city-block-wise through the space, I found through a considerable amount of experimentation that it is the most efficient way to proceed.

To guard against bad local maxima ($\alpha$=0 in s orthogonal directions), multiple starting points for the $\mathbf{X_i}^{(1)}$'s are utilized. If different solutions are found (which are rare and almost always close together) then the lines joining the unique local maxima are searched for the best solution. After considerable experimentation, I found that three starting points worked very well in practice. One starting point is from the eigenvalue-eigenvector decomposition of the double-centered agreement score matrix (see Chapter 2) and the other two are randomly generated.

In Poole (2000) I show extensive Monte Carlo studies of the legislator procedure using perfect as well as data with error.[6] With perfect data the legislator procedure almost always finds the legislator polytope that perfectly classifies the legislator's choices. When error is present the recovery of the legislator points is very good. Not surprisingly, as the number of cutting planes increases with the error level held fixed, the precision of the recovery of the legislators increases dramatically.

***Overall OC Algorithm***

The OC algorithm consists of three phases:

1) Generate starting values for the legislators, the $\mathbf{X_i}$'s, from an eigenvalue/eigenvector decomposition of the legislator by legislator agreement score matrix.
2) Given the $\mathbf{X_i}$'s, find the optimal estimates of the normal vectors, the q $\mathbf{N_j}$'s.
3) Given the $\mathbf{N_j}$'s, find the optimal estimates of the $\mathbf{X_i}$'s.
4) Go to (2).

With error in one dimension steps (2) to (4) are the Edith algorithm. In two or more dimensions step (2) is the cutting plane procedure and step (3) is the legislator procedure. In Poole (2000) I report an extensive set of Monte-Carlo experiments applying the OC algorithm to perfect data and data with error.[7] With perfect data I found that the algorithm works well regardless of the number of dimensions. OC also works reasonably well when the dimensions are not equally salient. For example, in two dimensions if 85 percent of the cutting lines are nearly parallel to the second dimension, the legislator configuration is recovered with reasonable precision. However, in real world applications where noise is present, such data will look like it fits a one-dimension model.

Consequently, there is no substitute for the researcher's substantive understanding of the data.

Given the history of other multidimensional scaling techniques, most empirical applications of OC will be to data matrices with missing entries and the estimated configurations will be in three or fewer dimensions. Missing data presents no problem for the algorithm. In the cutting plane procedure it simply means that the total number of legislators may vary from vote to vote. In the legislator procedure it simply means that the number of cutting lines may vary from legislator to legislator. Handling missing data requires a little bookkeeping but it has no effect on the algorithm. OC works very well with and without error at high levels of missing data.

### ***Conclusion***

OC is a general non-parametric unfolding technique for maximizing the correct classification of binary preferential choice data. The motivation for and the primary focus of OC is parliamentary roll call voting data but the procedures that implement the unfolding can also be applied to a variety of other problems. In particular, the cutting plane procedure avoids the pitfalls of probit and logit when they are applied to roll calls on which there is perfect classification.

Although neither the cutting plane nor the legislative procedure can be formally shown to converge to the global classification maximum, Monte-Carlo tests reported in Poole (2000) show that both in fact work very well in practice. In the presence of error the cutting plane procedure almost certainly passes through or very near to the classification maximum and the maximum can be recovered from the iteration record. The legislative procedure is guaranteed to converge to a very strong local maximum. That is, a local maximum for which the point cannot be moved in any orthogonal direction and have the correct classifications increase. When the two procedures are used together in an alternating framework to analyze binary choice matrices, their performance is excellent. The Monte Carlo tests reported in Poole (2000) are testimony to this fact.

In Chapter 4 I discuss probabilistic models of legislative voting using the geometry of Chapter 2. These models will build upon the OC framework because OC in turn is built upon the geometry. This lends stability to the parametric models that they otherwise would not have.

# Appendix

The following two well known matrix decomposition theorems are an important part of the scaling programs discussed in this book. Theorem I states that every *rectangular* matrix of real numbers can be written as the product of two orthogonal matrices and one diagonal matrix. This is known as *singular value decomposition*. Theorem I was stated by Eckart and Young (1936) in their famous paper but they did not provide a proof. The first proof of Theorem I was given by Johnson (1963). Horst (1963) refers to the decomposition shown in Theorem I as the *basic structure* of a matrix and discusses the mechanics of matrix decomposition in detail in chapters 17 and 18. A more recent treatment can be found in chapters 1 and 2 of Lawson and Hanson (1974).

Theorem II – the famous Eckart-Young (1936) Theorem – solves the general least squares problem of approximating one matrix by another of lower rank. Geometrically, suppose the matrix is a set of p points in an n dimensional space and we wish to find the best two dimensional plane through the p points such that the distances from the points to the surface of the plane are minimized. Technically, let A be a p by n matrix of rank 15 and let B be a p by n matrix of rank 2. Given A, the problem is to find the matrix B such that $\sum_{i=1}^{p}\sum_{j=1}^{n}\left(a_{ij}-b_{ij}\right)^2$ is minimized.

Theorem II was never explicitly stated by Eckart and Young. Rather, they use two theorems from linear algebra (Theorem I was the first) and a very clever argument to show the truth of their result. Later, Keller (1962) independently rediscovered the Eckart-Young result (Theorem II).

**Theorem I (Singular Value Decomposition)**

Let **A** be a p by n matrix of real elements (not all zeroes) with $p \geq n$. Then there is a p by p orthogonal matrix **U**, an n by n orthogonal matrix **V**, and a p by n matrix $\Lambda$ such that

$$\mathbf{A} = \mathbf{U}\Lambda\mathbf{V}' \qquad \text{and} \qquad \mathbf{U}'\mathbf{A}\mathbf{V} = \Lambda$$

where

$$\Lambda = \left[\frac{\Lambda_n}{\mathbf{0}}\right]$$

and $\mathbf{U}'\mathbf{U} = \mathbf{U}\mathbf{U}' = \mathbf{I}_p$, $\mathbf{V}'\mathbf{V} = \mathbf{V}\mathbf{V}' = \mathbf{I}_n$, where $\mathbf{I}_p$ and $\mathbf{I}_n$ are p by p and n by n identity matrices respectively. $\Lambda_n$ is an n by n diagonal matrix and **0** is a p-n by n matrix of zeroes. The diagonal entries of $\Lambda_n$ are non negative with exactly s entries strictly positive ($s \leq n$).

**Theorem II (Eckart and Young)**

Given a p by n matrix **A** of rank $r \leq n \leq p$, and its singular value decomposition, $\mathbf{U}\Lambda\mathbf{V}'$, with the singular values arranged in decreasing sequence

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \lambda_n \geq 0$$

then there exists a p by n matrix **B** of rank s, $s \leq r$, which minimizes the sum of the squared error between the elements of **A** and the corresponding elements of **B** when

$$\mathbf{B} = \mathbf{U}\Lambda_s\mathbf{V}'$$

where the diagonal elements of $\Lambda_s$ are

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \lambda_s > \lambda_{s+1} = \lambda_{s+2} = \ldots = \lambda_n = 0$$

Theorem I states that every real matrix can be written as the product of two orthogonal matrices and one diagonal matrix. Theorem II states that the least squares approximation in s dimensions of a matrix **A** can be found by replacing the smallest n-s roots of Λ with zeroes and remultiplying $\mathbf{U\Lambda V'}$.

Because the lower p-n rows of Λ are all zeros, it is convenient to discard them and work only with the n by n diagonal matrix $\Lambda_n$ . In addition, the p-n eigenvectors in **U** corresponding to the p-n lower rows of Λ may also be discarded. With these deletions of redundant rows and columns, **U** is a p by n matrix, Λ is an n by n diagonal matrix, and **V** is an n by n matrix. Hence, $\mathbf{U'U = V'V = VV' = I_n}$. A decomposition according to Theorem I will be assumed to be in this form.

# Chapter 3 Notes

---

[1] Personal communication to the author from Willem J. Heiser, 27 March 1998. Some recent examples of probabilistic/metric models within the psychometrics tradition are Heiser (1981); DeSarbo and Hoffman (1987); Gifi (1990); Blokland-Vogelesang (1991); Hojo (1994); and Andrich (1995).

[2] This is not as far-fetched as it sounds. Several European parliaments classify at 95% or above in one or two dimensions. For example, most legislative sessions during the French 4th Republic (Rosenthal and Voeten, 2004), recent sessions of the Czech parliament (personal communication from Abdul Noury), and the 1841 English parliament (personal communication from Cheryl Schonhardt-Bailey).

[3] This point is discussed in greater detail in a technical appendix to this chapter that can be found on the website for this book: http://voteview.uh.edu/spatialbook.htm.

[4] Additional Monte Carlo results are reported in a refereed but unpublished Appendix at http://voteview.uh.edu/paapp/paapp.htm.

[5] The first two dimensions estimated by NOMINATE classify about 85 percent of the roll call choices during the post World War II period (Poole and Rosenthal ,1997, ch. 2).

[6] See note (4) above.

[7] See note (4) above.