opscale: A Function for Optimal Scaling

by William G. Jacoby

Abstract The opsacle function from the optiscale package can be used to perform an optimal scaling transformation on a qualitative data vector, relative to a quantitative vector. In this paper, I provide a brief overview of optimal scaling, and an explanation of the opsacle function. opsacle is intended to be used with other R functions to create alternating least squares, optimal scaling (ALSOS) data analysis routines. I present a simple example in which opsacle is used along with the lm function to carry out an ordinal regression analysis on data from a public opinion survey.

Introduction

The opscale function performs an optimal scaling transformation on a data vector. The vector usually contains values of a qualitative variable to which a statistical model has been fitted. The optimal scaling transformation is applied to the qualitative variable, relative to a set of (quantitative) predicted values obtained from the statistical model. The resultant optimally scaled scores are the numerical values that maximize the correspondence between the empirical observations and the statistical model, subject to constraints implied by measurement assumptions about the variable.

The general idea of optimal scaling can be traced back at least to R. A. Fisher, and it has appeared in a number of different contexts in statistics and data analysis. The particular motivation behind the opscale function is to facilitate the development of alternating least squares, optimal scaling (ALSOS) routines for the quantitative analysis of qualitative data (Young, 1981) within the R statistical computing environment. This manuscript will provide an overview of optimal scaling, an explanation of the opscale function, and an example illustrating how the function can be used to perform an ALSOS regression analysis of data that are assumed to exist at the ordinal level of measurement.

An overview of optimal scaling

Let $x$ be a qualitative vector of observations, with elements $x_1, x_2, \ldots, x_n$. The elements of $x$ need not be numeric although, in practice, they usually are. Along with the observation vector, there is also a vector of quantitative values, say $y$, with elements $y_1, y_2, \ldots, y_n$. The elements of $y$ have a one-to-one correspondence with the elements of $x$; that is, $x_i$ corresponds to $y_i$, $x_2$ corresponds to $y_2$, and so on. Let $x^*$ be the vector of optimally scaled values of $x$, with respect to $y$. The $x^*$ vector is defined as a set of numeric values, $x^*_1, x^*_2, \ldots, x^*_n$, that is (1) maximally correlated with the entries in $y$, while (2) preserving the measurement characteristics that are assumed for $x$. Optimal scaling is the procedure for obtaining $x^*$ from $x$, $y$, and the measurement assumptions that the analyst makes about $x$.

There are three measurement characteristics that are relevant to an optimal scaling transformation. First, measurement level can be either nominal or ordinal, indicating whether the observational categories of $x$ are merely identity-preserving or are ordered according to some characteristic of the observations. Second, measurement process can be discrete or continuous, indicating whether observations within a given category of $x$ must be assigned the same optimally scaled value in $x^*$, or can take on different values within a closed interval. Third, measurement conditionality divides a dataset into partitions, within which observations and scores can be compared meaningfully to each other. In the present context, we will assume that the entire $x$ vector comprises a single partition.

Empirical data can exist with any combination of measurement characteristics. So, we need to consider the procedures for optimal scaling under the four situations that arise when measurement level and measurement process are crossed against each other. First, the discrete-nominal case imposes only the following measurement restriction on the entries in the vector of optimally scaled values:

$$x_i = x_j \Rightarrow x^*_i = x^*_j$$  \hspace{1cm} (1)

The optimal scaling procedure simply takes the conditional means of the $y_i$'s within the observational categories of $x$, and assigns those means to the entries in $x^*$ corresponding to the respective categories. Young (1981) points out that this is equivalent to Fisher’s appropriate scoring technique (Fisher, 1938). The $x^*$ vector provides least-squares fit to $y$ while still maintaining condition (1).

Second, the discrete-ordinal case uses the following restrictions for the optimally scaled values:

$$x_i = x_j \Rightarrow x^*_i = x^*_j$$  \hspace{1cm} (2A)

$$x_i < x_j \Rightarrow x^*_i \leq x^*_j$$  \hspace{1cm} (2B)

The optimal scaling procedure begins by calculating the conditional $y$ means, as in the discrete-nominal

$^1$An optimal scaling transformation could also be carried out on data measured at the interval or ratio level. But, it is not particularly interesting or useful to do so, because only linear transformations are typically permitted at those measurement levels.
case. After doing so, Kruskal’s monotonic transformation with the secondary treatment of tied values is applied to the conditional means in order to obtain the values in $x^*$ (Kruskal, 1964). Here, the $x^*$ vector is least-squares with respect to $y$ but also weakly monotonic with respect to $x$.

Turning to the continuous measurement process, conditions (1) and (2A) are replaced by the following:

$$ x_i = x_j \Rightarrow x^*_{\text{max,lower}} \leq x^*_i, x^*_j \leq x^*_{\text{min,higher}} $$

(3)

Where $x^*_{\text{max,lower}}$ refers to the largest entry in $x^*$ that is assigned to any $x_k \neq x_i, x_j$, but that is still smaller than either $x^*_i$ or $x^*_j$. Similarly, $x^*_{\text{min,higher}}$ refers to the smallest entry in $x^*$ that is assigned to any $x_k \neq x_i, x_j$, but that is still larger than either $x^*_i$ or $x^*_j$. All of this is a somewhat clumsy way of saying that empirical categories from $x$ now correspond to intervals of real numbers in $x^*$, rather than to single numerical values.

The optimal scaling transformation for continuous-ordinal data is performed by carrying out Kruskal’s monotonic transformation with the primary treatment of tied values. Furthermore, the transformation is applied to the individual $y_i$’s, rather than to the conditional means. The resultant $x^*$ is a least-squares fit to $y$, while consisting of intervals of values that are weakly monotonic with respect to the observational categories of $x$.

For the continuous-nominal case, we will use the two-step “pseudo-ordinal” procedure proposed by de Leeuw, Young, and Takane (1976). In the first step, the data are treated as nominal-discrete to establish an ordering of the categories in $x$, using the $x^*_i$ values from this step. Then, in the second step, the categories are treated as ordinal-continuous to obtain the interval of optimally scaled values for each of the $x_i$’s. de Leeuw et al. (1976) and Young (1981) both caution that the $x^*$ obtained from this procedure will not necessarily be globally optimal. But, it is a least-squares fit to the specific ordering of the categories in $x$ obtained on the first step, and that is usually adequate in practical applications.

Optimal scaling can be viewed as a strategy for assigning a particular set of numbers to objects in a non-arbitrary manner. In most optimal scaling applications, $y$ is a vector of predicted values from a statistical model that has been fitted to the elements of $x$. Thus, it is reasonable to say that the optimally scaled values for the objects in $x^*$ are those numbers that, taken together with the model parameter estimates, maximize the fit to a statistical model containing $x$. The advantage of the optimal scaling approach is that it incorporates empirical information along with a priori measurement assumptions about the substantive differences between the objects being measured. Young (1987) argues that the OS strategy effectively treats measurement properties as testable hypotheses about the observations.

The opscale function

In R, optimal scaling can be carried out with the opscale function from the optiscale package. The only required argument for opscale is x.qual, the name of a qualitative vector. The main optional arguments are: x.quant, the name of a quantitative vector (if omitted, a vector of sequential integers is used); level, the measurement level of x.qual with 1= nominal (the default) and 2=ordinal; and process, the measurement process of x.qual with 1=discrete (the default) and 2=continuous.

The opscale function produces an object of class “opscale”. An opscale object is a list containing (among other elements) a vector named os that is an optimally scaled version of x.qual relative to x.quant. The print method for opscale objects lists the elements of x.qual, x.quant, and os. The summary method matches the original qualitative values with their corresponding optimally-scaled values, and the plot method produces a line plot of the latter versus the former. Additional methods for opscale objects produce a Shepard diagram (using the shepard function) and Kruskal’s Stress coefficients (using the stress function).

The following R code shows the main results produced when the opscale function is applied to some hypothetical data, under varying measurement assumptions. Note that rescale=TRUE is the default, and it would have rescaled the os object to the same mean and standard deviation as $x_1$. Setting this argument to FALSE makes it easier to see how the optimally-scaled values are related to the elements of $x_1$ and $x_2$:

```r
> # x1 is a vector of qualitative data
> # x2 is a vector of quantitative values
> x1 <- c(1,1,1,1,1,2,2,2,2,2,3,3,3,3)
> x2 <- c(3,2,2,2,2,1,2,3,4,5,2,6,5,6)
> # Optimal scaling, specifying that x1
> # is nominal-discrete
> op.scaled <- opscale(x.qual=x1, x.quant=x2,
+ level=1, process=1, rescale = FALSE)
> op.scaled$os
[1] 2.25 2.25 2.25 2.25 2.00 2.00 2.00 4.50
[9] 4.50 4.50 4.50 4.50 4.50
> # Specify that x1 is ordinal-discrete
> op.scaled2 <- opscale(x.qual=x1, x.quant=x2,
+ level=2, process=1, rescale = FALSE)
> op.scaled2$os
[1] 2.142857 2.142857 2.142857 2.142857
[13] 4.500000
> # Specify that x1 is ordinal continuous
> op.scaled3 <- opscale(x.qual=x1, x.quant=x2,
```
The `opscale` function is similar to the `opscaal` function in SAS/IML. However, the latter only allows the user to specify the measurement level of a qualitative vector; the measurement process is always assumed to be discrete. The `opscale` function is more general because it allows the user to set the measurement process as well as the measurement level.

There are other ways to perform optimal scaling in R. For example, the `homals` package (de Leeuw and Mair, 2009) and the `aspect` package (Mair and de Leeuw, 2010) both perform optimal scaling. But, in each case, the scaling procedure is incorporated into a broader system—the Gifi classification of non-linear multivariate analyses in the former, and the framework of a multivariable combined with user-specified aspects in the latter. Thus, even though both of these packages give users a great deal of flexibility, optimal scores are only available for the data analysis procedures that are included within them.

From a slightly different perspective, optimal scores are an important element of correspondence analysis (CA), which can be carried out with several R functions (Mair and Hatzinger, 2012). CA generates scores for observational categories that maximize the correlation across the ways of the input data matrix. Similarly, optimal scaling is used in nonmetric multidimensional scaling (MDS), available through several R functions including `sammon` and `isoMDS` in the `MASS` package, and `smacofSym` and `smacofIndDiff` in `smacof` (Mair and de Leeuw, 2009). Specifically, the disparities used to obtain the MDS solution are optimally scaled transformations of the input ordinal dissimilarities, relative to the distances in the scaling solution.

All of the preceding applications employ optimal scaling in specific contexts; that is, they provide optimally-scaled values as a function of some data analytic model. In contrast, `opscale` is a stand-alone function. Therefore, it can be used as a utility tool to incorporate an optimal scaling step in new procedures, or to adapt existing statistical models in ways that enable them to accommodate qualitative data.

### An example: ordinal regression via the MORALS algorithm

The `opscale` function can be used to create ALSOS routines for quantitative analysis of qualitative data. An ALSOS analysis always consists of the following steps:

1. The variables are assigned initial optimal scale values, and the measurement characteristics are set.
2. Least-squares estimates are obtained for the parameters of the statistical model.
3. If model fit has not improved over the previous iteration, terminate the procedure; otherwise proceed with the following steps.
4. The predicted values from the statistical model are used to generate new optimal scale values for the variables.
5. Return to Step 2 and re-estimate the model using the updated optimally-scaled variable values.

A wide variety of data analysis situations can be handled simply by substituting the appropriate statistical model in step 2. One specific manifestation of ALSOS, the MORALS (Multiple Optimal Regression via Alternating Least Squares) algorithm, can be used to fit a multiple regression model to ordinal dependent and independent variables (Young, de Leeuw, and Takane, 1976).

Note that MORALS is very similar to the ACE (Alternating Conditional Expectations) algorithm developed by Breiman and Friedman (1985) and available in R through the `ace` function in the `acepack` package. Both MORALS and ACE seek the “best” transformations for the variables in a multiple regression model; that is, nonlinear functions of the original variables that maximize the model’s fit to the data. The difference is that MORALS obtains the transformations through optimal scaling while ACE uses a smoother.

The current example will replicate an analysis reported by Jacoby (1999). The data are taken from the Center for Political Studies’ 1992 National Election Study (NES), a national public opinion survey conducted from September through November, 1992. The dependent variable is a measure of relative candidate preference, defined as each survey respondent’s “feeling thermometer” rating of George H. W. Bush minus his or her feeling thermometer rating of Bill Clinton. Feeling thermometers are, basically, 0-100 rating scales, where scores of 50 are neutral, scores above 50 indicate the respondent feels warmer toward the candidate, while scores below 50 indicate the respondent feels cooler toward the candidate. So, the choice variable ranges from -100 for a person who

```R
+ level=2, process=2, rescale = FALSE)
> op.scaled3$os
[1] 2.0 2.0 2.0 2.0 2.0 2.5 4.0 5.0 2.5 6.0
> # Specify that x1 is nominal continuous
> op.scaled4 <- opscale(x.qual=x1, x.quant=x2,
+ level=1, process=2, rescale = FALSE)
>
> # Specify that x1 is nominal continuous
> op.scaled4$os
[1] 2.0 2.0 2.0 2.0 2.0 2.5 4.0 5.0 2.5 6.0

A wide variety of data analysis situations can be handled simply by substituting the appropriate statistical model in step 2. One specific manifestation of ALSOS, the MORALS (Multiple Optimal Regression via Alternating Least Squares) algorithm, can be used to fit a multiple regression model to ordinal dependent and independent variables (Young, de Leeuw, and Takane, 1976).

Note that MORALS is very similar to the ACE (Alternating Conditional Expectations) algorithm developed by Breiman and Friedman (1985) and available in R through the `ace` function in the `acepack` package. Both MORALS and ACE seek the “best” transformations for the variables in a multiple regression model; that is, nonlinear functions of the original variables that maximize the model’s fit to the data. The difference is that MORALS obtains the transformations through optimal scaling while ACE uses a smoother.

The current example will replicate an analysis reported by Jacoby (1999). The data are taken from the Center for Political Studies’ 1992 National Election Study (NES), a national public opinion survey conducted from September through November, 1992. The dependent variable is a measure of relative candidate preference, defined as each survey respondent’s “feeling thermometer” rating of George H. W. Bush minus his or her feeling thermometer rating of Bill Clinton. Feeling thermometers are, basically, 0-100 rating scales, where scores of 50 are neutral, scores above 50 indicate the respondent feels warmer toward the candidate, while scores below 50 indicate the respondent feels cooler toward the candidate. So, the choice variable ranges from -100 for a person who
is maximally favorable toward Clinton, and maximally negative toward Bush, through zero for someone who feels neutral toward the two candidates, to +100 for someone who is very warm toward Bush, but very cool about Clinton.

There are three independent variables: Party identification; ideological self-placement, and assessments about whether the national economy was getting better or worse over the previous year. Table 1 shows the categories for each of these variables, along with the numeric scores assigned in the NES codebook.

Table 1: Category scores for independent variables in MORALS example.

<table>
<thead>
<tr>
<th>Party identification:</th>
<th>0</th>
<th>Strong Democrat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Democrat</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Independent leaning Democratic</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Independent</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Independent leaning Republican</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Republican</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Strong Republican</td>
</tr>
</tbody>
</table>

| Ideological self-placement: | 1 | Extremely liberal |
|                            | 2 | Liberal |
|                            | 3 | Slightly liberal |
|                            | 4 | Moderate, middle of the road |
|                            | 5 | Slightly conservative |
|                            | 6 | Conservative |
|                            | 7 | Extremely conservative |

| Did nation’s economy become better or worse over past four years? | 1 | Much better |
|                                                               | 2 | Better |
|                                                               | 3 | Stayed the same |
|                                                               | 4 | Worse |
|                                                               | 5 | Much worse |

Let us consider the measurement levels of the variables included in the model. Feeling thermometers are intended to provide relatively continuous, interval-level assessments of the presidential candidates. But, most survey respondents report thermometer scores that end in zero, effectively rendering them into discrete, eleven-category scales. Similarly, most social scientists probably would say that all three of the independent variables are measured at the ordinal level. But, they are all routinely used in regression models, suggesting that they actually are treated as interval-level for practical purposes.

An ALSOS analysis can be viewed as a diagnostic test to determine whether or not it is reasonable to treat these variables as interval-level measurement. The logic is straightforward: We estimate the regression parameters and optimal scores for the variables using the MORALS algorithm, specifying that all of the variables are measured at the ordinal level. If the resultant optimal scores for a variable are linearly related to the original scores, then nothing is lost by treated the variable as interval-level measurement.

Assume that we will carry out a “quick and dirty” descriptive MORALS analysis in R, with code that is aimed at clarity and exposition rather than efficiency. The complete R session to carry out the analysis can be found in the optiscale package vignette. The session begins by loading optiscale and retrieving the NES data in the data frame elec92 (included with the package). Then we initialize various objects to control the ALSO iterations: a counter (niter); the fit from the previous iteration (previous.rsquared); a measure of change in model fit (r_squared.differ); and an iteration history (record). The optimally-scaled versions of the dependent variable (dvar.os) and the three independent variables (party.os, ideol.os, and econ4yr.os) are all initialized to the category values assigned in the NES codebook.

The MORALS analysis, itself, is carried out by the program loop shown in Figure 1. In the figure, the first five lines of code estimate the regression model, assign the results to an object (reg.os), and update the iteration history. If the model fit has not converged, the loop goes on to obtain predicted values for each variable, perform the optimal scaling, and assign the new optimal scores. Note that the optimal scaling step must be carried out sequentially for each of the variables. The OS transformations cannot be performed with a function that is more efficient from a computational perspective (e.g., mapply) because, with correlated data, the optimal scores for any given variable are affected by the scores assigned to all other variables. After running the ALSOS loop, the iteration history can be produced as follows:

```
> record <- matrix(data=record, ncol=3, byrow=T)
> colnames(record) <- c("Iteration", 
> "R-squared", "Improvement")
> rownames(record) <- rep("", nrow(record))
>
> record

Iteration R-squared Improvement
1 0.5411882 0.5411882247
2 0.5943073 0.0531191133
3 0.5946870 0.0003796453
```

The optimal transformations on the variables increased the model fit from 0.541 to 0.595, and virtually all of the change occurred on the first iterate.

---

2Note that, if there were questions about the proper ordering of the categories on any of these variables, then the relevant variables could be treated as nominal-level. The optimal scores would then provide empirical information about the “best” category ordering, at least with respect to this particular regression model.

3An inferential analysis involves additional complications because the degrees of freedom associated with the model must also take the optimal scoring process into account.
while (rsquared.differ > .001 && niter <= 30) {

niter <- niter + 1
reg.os <- lm(dvar.os ~ party.os + ideol.os + econ4yr.os)
rsquared.differ <- summary(reg.os)$r.squared - previous.rsquared
previous.rsquared <- summary(reg.os)$r.squared
record <- c(record, niter, summary(reg.os)$r.squared, rsquared.differ)

if (rsquared.differ > .001) {
  dvar.pred <- predict(reg.os)
  opscaled.dvar <- opscale(elec92$choice, dvar.pred, level=2)
dvar.os <- opscaled.dvar$os

  party.pred <- (dvar.os - (reg.os$coefficients[1] +
    (reg.os$coefficients[3]*ideol.os) +
    (reg.os$coefficients[4]*econ4yr.os))) * (1/reg.os$coefficients[2])
  opscaled.party <- opscale(elec92$party, party.pred, level=2)
party.os <- opscaled.party$os

  ideol.pred <- (dvar.os - (reg.os$coefficients[1] +
    (reg.os$coefficients[2]*party.os) +
    (reg.os$coefficients[4]*econ4yr.os))) * (1/reg.os$coefficients[3])
  opscaled.ideol <- opscale(elec92$ideol, ideol.pred, level=2)
  ideol.os <- opscaled.ideol$os

  econ4yr.pred <- (dvar.os - (reg.os$coefficients[1] +
    (reg.os$coefficients[3]*ideol.os) +
    (reg.os$coefficients[2]*party.os))) * (1/reg.os$coefficients[4])
  opscaled.econ4yr <- opscale(elec92$econ4yr, econ4yr.pred, level=2)
econ4yr.os <- opscaled.econ4yr$os
}
}

Figure 1: ALSOS loop for carrying out MORALS analysis of NES data.

tion. This rapid convergence with relatively small improvement is fairly typical for MORALS analyses with ordinal, discrete variables.

The regression intercept and the coefficients for the optimally-scaled independent variables can be obtained by printing out the relevant elements from the reg.os object (once again, the default standard errors and significance tests would be incorrect):

> summary(reg.os)$coefficients[,1]
   (Intercept)  party.os  ideol.os  econ4yr.os
   -28.967305  12.838867   5.580363  -7.841443

Recall that the independent variables are coded so that larger values correspond to stronger Republican (or less Democratic) identifications, more conservative (or less liberal) self-placements, and beliefs that the economy got worse. So the signs of the regression coefficients show that Republicans, conservatives, and those who thought the economy was getting better all rated Bush higher relative to Clinton and vice versa. All of this is fully consistent with prior expectations and other empirical analyses of voting behavior during the 1992 presidential election (e.g., Alvarez and Nagler, 1995).

Perhaps the most interesting results from the MORALS algorithm involve the optimal scaling of the four variables in the model. If the variables really are measured at the interval level, then the optimally scaled scores should be linear functions of the original scores. One way to examine the optimal transformations is to use the summary function for an opscale object. For example:

> summary(opscaled.party)

<table>
<thead>
<tr>
<th>Measurement Level: Ordinal</th>
<th>Measurement Process: Discrete</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial.values OR values</td>
<td>1  0 0.2191755</td>
</tr>
<tr>
<td></td>
<td>2  1 1.3103895</td>
</tr>
<tr>
<td></td>
<td>3  2 1.3103895</td>
</tr>
<tr>
<td></td>
<td>4  3 2.6572240</td>
</tr>
<tr>
<td></td>
<td>5  4 4.3736945</td>
</tr>
<tr>
<td></td>
<td>6  5 4.7849966</td>
</tr>
<tr>
<td></td>
<td>7  6 6.1690615</td>
</tr>
</tbody>
</table>

While this gives the precise original and optimally scaled scores, it is probably easier to get a sense of the optimal transformation by plotting the latter against the former. This can be accomplished very easily, using the plot method for opscale objects:

plot(reg.os$dep.var)
plot(reg.os$party.os)
The resulting graphical displays are shown in Figure 2. Three of the four plots show fairly obvious nonlinearities. The dependent variable (plotted in the upper left panel in Figure 2) shows a roughly sigmoid shape, suggesting that the differences across the more extreme categories are less pronounced than are those closer to the center of the variable’s range. With party identification (the upper right panel), the optimally scaled differences between the categories originally coded 1 and 2, and (to a lesser extent) between those coded 4 and 5, are smaller than the differences between other adjacent categories. This means that independents who lean toward one of the parties and non-strong party identifiers are relatively similar to each other—especially on the Democratic side. The plot for economic assessments (the lower right panel in Figure 2) flattens out on the left side. This indicates that people who said the economy had gotten much better were no different than those who said that the economy had gotten somewhat better, in terms of their relationships with the other variables in the regression model. Only the ideological self-placement variable shows a nearly linear relationship between the optimally-scaled scores and the original NES codes. Taken together, these results suggest that the usual practice of treating feeling thermometers, party identification, and judgments about the American economy as interval level variables may, in fact, be problematic.

Conclusion

This paper introduces the `opscale` function for obtaining an optimal scaling transformation within the R statistical computing environment. Optimal scaling is, itself, described as “a data analysis technique which assigns numerical values to observation categories in a way which maximizes the relation between the observations and the data analysis model while respecting the measurement character of the data (Young, 1981, pg. 358). The `opscale` function is a utility tool typically used in conjunction with other R functions that estimate statistical models, enabling easy implementation of the ALSOS strategy for quantitative analysis of qualitative data. On the one hand, this should prove useful for the development of new nonmetric scaling procedures. On the other hand, and as the example shown above demonstrates, ALSOS can be used to combine optimal scaling with existing statistical models in useful ways. As Young (1981, pg. 358) says, “If a procedure is known for obtaining a least squares description of
numerical (interval or ratio measurement level) data then an ALSOS algorithm can be constructed to obtain a least squares description of qualitative data (having a variety of measurement characteristics).”

Thus, the `opscale` function should facilitate the analysis of data which are often viewed as problematic with respect to many statistical methodologies. This should be particularly useful for fields in which such data abound, such as the social and behavioral sciences.

Bibliography


William G. Jacoby
Michigan State University
Department of Political Science
East Lansing, MI 48824
jacoby@msu.edu

William G. Jacoby
Michigan State University
Department of Political Science
East Lansing, MI 48824
jacoby@msu.edu