

Recovering a Basic Space in R

January 13, 2011

1 Introduction

This R (R Development Core Team, 2010) package contains software designed to recover a latent dimension (i.e. a basic space) from a numeric matrix of data. The initial development was conducted by Poole (1998), who wrote the Fortran executable that was previously used by social scientists to analyze survey data in publications such as Saiegh (2009) and Abrajano and Poole (2010). The principal purpose of the package is to facilitate the analysis of self-placement and/or perceptual survey data by scaling stimuli into a common space. Stated differently, assume that various political actors (i.e. the stimuli) occupy positions on a continuous left-right political spectrum, and our objective is to recover estimates of those positions. This package facilitates the estimation of these positions under the assumption that survey respondents report their personal estimates of these positions with different levels of bias and scale attenuation. While intended for use with survey data, this procedure has potentially broader applications outside of social science because mathematically it can be thought of as a special case of singular value decomposition on a numeric matrix with missing data.¹

This vignette proceeds in four steps. First, we begin with a description of the math that underlies the basic space estimator. We then provide three examples. First, we show a Monte Carlo analysis that suggests our estimator produces an accurate decomposition of our simulated data matrix even with 30 per cent of the data missing. Secondly, we show how the procedure can be applied to self-placement survey data from the 1980 National Election Survey. We conclude with an application of the model to perceptual data from the 1980 National Election Study, where various political candidates are ranked along a 7 point liberal-conservative scale.

¹When used to analyze survey data, this property implies that survey scales can be treated as a continuous rather than ordinal variable. Researchers who believe that their scales are ordinal should therefore not be using the model presented here.

2 Model

The exposition of the model presented here closely follows Poole (1998). Consider a matrix of survey data X_0 with n respondents and m issue scales, with individuals on the rows and issues on the columns. Some cells of the matrix X_0 are missing, and we let X denote the version of X_0 that has no missing data. In each cell x_{ij} , respondent i ($i=1, \dots, n$) reports their position on issue scale j ($j=1, \dots, m$), with some responses missing.² Now let Ψ_{ik} be the i th individual's position on the k th basic dimension ($k=1, \dots, s$), W be an m by s matrix of weights that map individual positions from the basic space to the issue dimensions, c be a vector of issue dimension intercept terms of length m , J_n by an n length vector of ones, and E_0 be error terms in the data matrix. The model that we seek to estimate is:

$$X_0 = [\Psi W' + J_n c']_0 + E_0 \quad (1)$$

Without loss of generality, we also assume that E_0 is drawn from a symmetric distribution with mean 0 and that the centroid of the basic space coordinates is at the origin (i.e. $J'_n \Psi = 0$). Substituting into the model equation, this implies that $J'_n [X - J_n c'] = 0'_m$, where $0'_m$ is an m length vector of zeroes. Then in the situation where X_0 has no missing data, the parameters of interest can all be recovered using singular value decomposition. To see why this is true, recall that for an n by m matrix of real elements with $n \geq m$, there exists an n by n orthogonal matrix U , an m by m orthogonal matrix V , and an m by m matrix Λ such that:

$$X = U \Lambda V' \quad (2)$$

where Λ is an m by m diagonal matrix of singular values.³ To solve [1], set c equal to the column means of X , or $c_j = \frac{\sum_{i=1}^n x_{ij}}{n} = \bar{x}_j$. Then using [2], the singular value decomposition of $X - J_n c'$ can be expressed as:

$$X - J_n c' = U \Lambda V' = \Psi W'$$

This implies that in the absence of missing data, one solution for Ψ and W is:

$$\Psi = U \Lambda^{0.5}$$

$$W = V \Lambda^{0.5}$$

with $\Lambda^{0.5}$ being a diagonal matrix where diagonal elements are the square roots of Λ . While other solutions to this problem exist, Eckart and Young (1936) have shown that the least squares approximation in s dimensions of a matrix A can be found by using only the first s singular values of A along the diagonal of Λ and remultiplying $U \Lambda V'$.

²The '0' subscript indicates that some elements are missing from the matrix.

³A more general form of this equation can be written in which Λ is instead an m by m matrix and U is n by m .

In the presence of missing data in data matrix X_0 , the use of singular value decomposition to solve for W and Ψ is no longer possible, and we instead estimate \hat{W} and $\hat{\Psi}$ using an alternating least squares (ALS) technique that is similar to the procedures used in Carroll and Chang (1970) and Takane, Young, and de Leeuw (1977). The objective function to be minimized is the sum of the squared deviations across all cells in A after the columns have been adjusted for column means, or:

$$\xi = \sum_{i=1}^n \sum_{j=1}^{m_i} \left\{ \left[\sum_{k=1}^s \Psi_{ik} W_{jk} \right] + c_j - x_{ij} \right\}^2$$

In minimizing this objective function, two constraints from the earlier analysis in the absence of missing data are applied. First, we exploit the fact that Ψ and W are orthogonal matrices, which implies that $\Psi' \Psi = W' W$.⁴ Secondly, following our earlier restriction that $J'_n [X - J_n c'] = 0'_m$, $J'_n U = J'_n \Psi = 0'_m$ as well. These restrictions produce the Lagrangian multiplier problem:

$$\mu = \xi = 2\gamma' [\Psi' J_n] + \text{tr}[\Phi(\Psi' \Psi - W' W)]$$

where Φ is a symmetric s by s matrix of Lagrangian multipliers and γ is an s length vector of Lagrangian multipliers. Since all Lagrangian multipliers are zero,⁵ the partial derivatives of ξ are:

$$\frac{\partial \mu}{\partial \Psi_{ik}} = 2 \sum_{j=1}^{m_i} \left[\left(\sum_{l=1}^s w_{jl} \psi_{jl} \right) + c_j - x_{ij} \right] w_{jk} \quad (3)$$

$$\frac{\partial \mu}{\partial w_{jk}} = 2 \sum_{i=1}^{n_j} \left[\left(\sum_{l=1}^s w_{jl} \psi_{jl} \right) + c_j - x_{ij} \right] \psi_{jk} \quad (4)$$

$$\frac{\partial \mu}{\partial c_j} = 2 \sum_{i=1}^{n_j} \left[\left(\sum_{l=1}^s w_{jl} \psi_{jl} \right) + c_j - x_{ij} \right] \quad (5)$$

Let W^* be an m_i by s matrix with appropriate rows corresponding to missing entries in X_0 removed, x_{0i} be the length m_i row of X_0 , and c_0 be the length m_i vector of constants corresponding to the elements of x_{0i} . Then if $W^{*'} W^*$ exists, the i th row of Ψ can be estimated by setting (3) to zero, collecting the s partial derivatives of the i th row of Ψ into a vector and solving for ψ_j as:

$$\hat{\psi}_i = (W^{*'} W^*)^{-1} W^{*'} [x_{0i} - c_0] \quad (6)$$

which can of course be estimated using ordinary least squares. Similarly, let $\Psi_j^* = [\Psi_0 | J_0]$ be an n_j by $s + 1$ matrix with the appropriate rows corresponding to missing data removed and bordered by ones, w_j be the s length vector of row j in W , c_j be the j th element of c ,

⁴More specifically, $\Psi' \Psi = \Lambda^{0.5} U' U \Lambda^{0.5} = \Lambda^{0.5} I_m \Lambda^{0.5} = \Lambda = W' W$.

⁵See Appendix A in Poole (1998) for a full proof that all Lagrangian multipliers are zero.

and x_{0j} be the j th column of X_0 . Then if $\Psi_j^{*'}\Psi_j^*$ exists, w_j and c_j can be jointly estimated by combining (4) and (5) as:

$$\frac{\hat{w}_j}{\hat{c}_j} = (\Psi_j^{*'}\Psi_j^*)^{-1}\Psi_j^{*'}x_{0j} \quad (7)$$

Equations (6) and (7) represent the core set of equations that are used to solve for \hat{W} , \hat{c} , and $\hat{\Psi}$. Once a set of starting values have been generated, (6) and (7) are iterated until convergence is complete. Generation of appropriate start values is conducted one dimension at a time, and a more detailed justification of the procedure can be found in Poole (1998). On the first dimension, start values are generated by using the following three equations:

$$\hat{c}_j = \frac{\sum_{i=1}^{n_j} x_{ij}}{n_j} = \bar{x}_j \quad (8)$$

$$w_{j1} = \text{diag}(\Gamma) \quad (9)$$

where Γ is an m by m diagonal matrix with diagonal elements either set to 1 or -1 that maximizes the number of positive elements in the m by m covariance matrix $\Gamma[X_0 - J_n c']'[X_0 - J_n c']\Gamma$. Γ is found by a simple iterative process similar to that used to speed eigenvector/eigenvalue decomposition (Poole, 1998). Given equations (8) and (9), starting values for ψ are:

$$\hat{\psi}_{i1} = \frac{\sum_{j=1}^{m_i} \hat{w}_{j1}(x_{ij} - \hat{c}_j)}{m_i} \quad (10)$$

If the number of dimensions to be estimated $s > 1$, start values for other dimensions can be generated on the start values generated from earlier dimensions simply by replacing the data matrix X_0 with the matrix of residuals E_{0s} in equations (9) and (10). However, no further estimation of start values for \hat{c} is required. The matrix of residuals to be used for generating start values on dimension s is:

$$E_{0s} = X_0 - \sum_{j=1}^s \hat{\Psi}_s \hat{w}_s' - J_n \hat{c}'$$

This residual matrix allows the generation of higher-dimension start values by iterating Γ to maximize the positive elements in E_{0s} . The starting values are now:

$$\hat{\psi}_{is} = \frac{\sum_{j=1}^{m_i} \hat{w}_{js} e_{1(s-1)ij}}{\sum_{j=1}^{m_j} \hat{w}_{js}^2} \quad (11)$$

$$\hat{w}_{js} = \frac{\sum_{i=1}^{n_j} \hat{\psi}_{is} e_{(s-1)ij}}{\sum_{i=1}^{n_j} \hat{\psi}_{is}^2} \quad (12)$$

Summarizing the preceding discussion in full, the basic space technique decomposes an n by m matrix X_0 with $n > m$ following equation (1). In the absence of missing data, this decomposition can be solved using singular value decomposition as in equation (2). The basic space technique can therefore be thought of as a generalization of singular value decomposition to matrices with missing data. Estimation of (1) proceeds in three steps. In the first stage, starting values on the first dimension are generated for \hat{c}_j , w_{j1} , and $\hat{\psi}_{i1}$ by iterating equations (8)-(10) until convergence. In the second stage, if the number of dimensions to be estimated $s > 1$, higher dimensional starting values for $\hat{\psi}_{is}$ and \hat{w}_{is} are generated dimension by dimension using equations (11)-(12). Finally, the starting values generated in the preceding two stages are improved by iterating equations (6)-(7) until convergence.

3 Monte Carlo Test

In this section, we present the first of three motivating examples. We begin with a Monte Carlo example that tests the basic space technique against simulated data. Four key variables should be set in each simulation: the number of respondents N (set here to $N = 1000$), the number of issue scales (also referred to as stimuli, and set here to $M = 20$), the number of explanatory dimensions (set here to $s = 2$), the fraction of observations that are missing (set here as 0.3), and the distribution of error terms (set here as random uniform draws from -0.5 to 0.5). These variables can be changed for other simulations, but the restriction that $N > M$ must be hold true. In cases where $M > N$ please refer to the third example that uses *blackbox_transpose()* and *aldrmck()*.

```
> set.seed(1231)
> library(basicspace)

## BASIC SPACE SCALING PACKAGE
## Copyright 2009 - 2011
## Keith Poole, Howard Rosenthal, Jeffrey Lewis, James Lo, and Royce Carroll
## Support provided by the U.S. National Science Foundation
## NSF Grant SES-0611974

> N <- 1000
> M <- 20
> s <- 2
> fraction.missing <- 0.3
> E <- matrix(runif(N * M, min = -0.5, max = 0.5),
+           nrow = N, ncol = M)
```

To generate the X matrix (i.e. the matrix in (1) before missing values are introduced), separately generate the matrices that produce the singular value decomposition of X following (2). Also generate the J_n and c vectors from (1). While X can be generated directly in one step, creating the components separately enjoys two significant advantages. First, recovery of the true values of Ψ and W is simplified. Secondly, the creation of Λ separately allows us to more easily tune the dimensionality of the matrix as desired.

```
> U <- matrix(runif(N * s), nrow = N, ncol = s)
> D <- diag(seq(from = 2.1, by = -0.2, length.out = s))
> V.prime <- matrix(runif(s * M), nrow = s, ncol = M)
> c <- rnorm(M)
> Jn <- rep(1, N)
```

With the intermediate matrices just generated, we can produce our X matrix by using equation (1) and the true Ψ and W matrices using: $\Psi = U\Lambda^{0.5}$ and $W = V\Lambda^{0.5}$.

```
> X.true <- U %*% D %*% V.prime + Jn %o% c
> X.0 <- X.true + E
> Psi.true <- U %*% sqrt(D)
> W.true <- t(V.prime) %*% sqrt(D)
```

X_0 is simply the X matrix with missing data values included completely at random, so we insert our missing data code (999 in the example) into the appropriate fraction of values as follows:

```
> missing <- sample(1:(N * M), round(fraction.missing *
+   N * M))
> X.0[missing] <- 999
```

The final step before estimation is to assign row and column names to the data set prior to input. In most applications these names are generally pulled from a survey, but they can also be generated manually:

```
> rownames(X.0) <- paste("Legis", 1:N, sep = "")
> colnames(X.0) <- paste("V", 1:M, sep = "")
```

Estimation of the Monte Carlo data after formatting is trivial. The function that applies the basic space decomposition described in this paper is *blackbox()*. It takes four arguments: the matrix to be decomposed, a vector of missing data values, a boolean flag indicating whether verbose output is desired, the number of dimensions to estimate, and the minimum number of issue scales that an individual needs to provide responses to if they are to be included in the estimation.

```
> result <- blackbox(X.0, missing = c(999), verbose = TRUE,
+   dims = 2, minscales = 8)
```

Beginning Blackbox Scaling...20 stimuli have been provided.

Blackbox estimation completed successfully.

```
> names(result)

[1] "stimuli"      "individuals" "fits"        "Nrow"
[5] "Ncol"         "Ndata"       "Nmiss"       "SS_mean"
[9] "dims"
```

The output object contains multiple data frames summarizing the results of the estimation. The key data frames are **stimuli**, which contain estimates of \hat{W} and c , as well as **individuals**, which contain estimates of $\hat{\Psi}$. The other quantities are fit statistics described in greater detail in the standard documentation for the function.

With the estimates complete, we are now able to test the recovery of our parameters of interest. In general, scaling problems are not fully identified. Stated differently, given $X = \Psi W'$, Ψ and W' are not unique solutions because $X = \Psi K K^{-1} W'$ for any conformable and invertible matrix K , so X can always be decomposed instead as $X = \Psi^* W'^*$ where $\Psi^* = \Psi K$ and $W'^* = K^{-1} W'$. When evaluating parameter fit, we are therefore largely concerned with finding monotonic relationships between the true and estimated parameters of interest. Figure 1 compares the true vs. estimated values of Ψ across two dimensions, and the results suggest a reasonable model fit.

Similarly, we can repeat the same procedure for W as well, a comparison shown in Figure 2. In Figure 3 we repeat this analysis for c , which is largely a column mean that is only estimated in one dimension. In both cases the estimates for \hat{W} and \hat{c} are a monotonic transformation of the true parameters as expected.⁶

Finally, we pool our estimates of \hat{W} , \hat{Psi} , and \hat{c} together to estimate the full matrix \hat{X} following Equation (1). While social scientists are principally concerned with estimation of \hat{W} and \hat{Psi} , others seeking to conduct singular value decomposition of matrices with missing data may find \hat{X} to be of value. One obvious application of \hat{X} is its potential use as an imputation tool for missing data.⁷ To test the viability of this idea, we separately plot the true values of X against the estimated values of \hat{X} separately for the cells retained in the estimation, and compared those results to estimates of \hat{X} in cells that were discarded prior to estimation to simulate the missing data mechanism. Figure 4 presents our results for retained vs. imputed X . What is particularly notable about this result is the close similarity between these plots — the imputed values not only appear reasonable (i.e. line up with the true values along a 45° line), but imputed values do not appear to have significantly higher mean squared error than the values that were retained (i.e. variance along the 45° line is

⁶In other estimates, the relationship may only be affine because $X = \Psi W'$ implies $X = -(\Psi) - (W')$ as well.

⁷The simulation presented here simulates missing data under the Missing Completely at Random (MCAR) assumption — nevertheless, there is no reason to think that this would not work under conditions where data are instead Missing at Random (MAR).

```

> par(mfrow = c(1, 2))
> plot(Psi.true[, 1], result$individuals[[2]]$c1,
+      xlim = c(0, 1.5), ylim = c(-0.65, 0.6), pch = 20,
+      cex = 0.4, cex.lab = 1.6, bty = "n", xlab = "True Psi, first dimension",
+      ylab = "Recovered Psi, first dimension")
> plot(Psi.true[, 2], result$individuals[[2]]$c2,
+      xlim = c(0, 1.5), ylim = c(-0.65, 0.6), pch = 20,
+      cex = 0.4, cex.lab = 1.6, bty = "n", xlab = "True Psi, second dimension",
+      ylab = "Recovered Psi, second dimension")

```

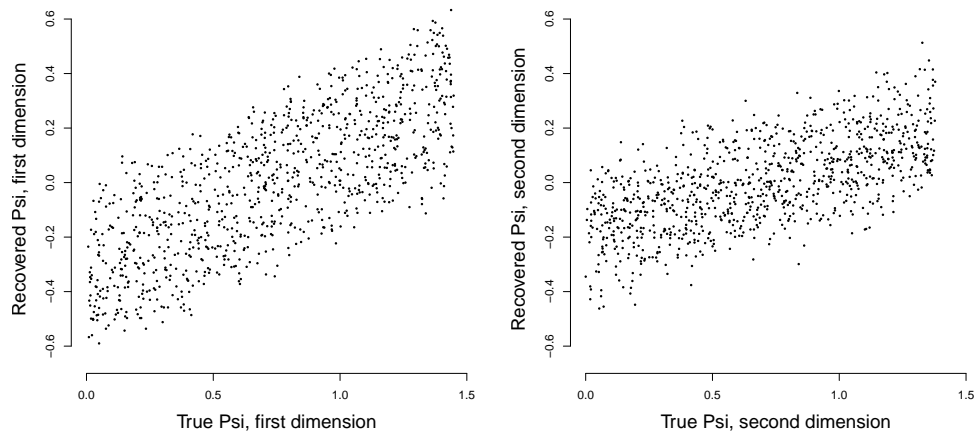


Figure 1: Plots of True vs. Estimated Ψ scores, first and second dimension.

```

> par(mfrow = c(1, 2))
> plot(W.true[, 1], result$stimuli[[2]]$w1, ylim = c(1,
+ 2.6), pch = 20, cex = 1.5, cex.lab = 1.6,
+ bty = "n", xlab = "True W, first dimension",
+ ylab = "Recovered W, first dimension")
> plot(W.true[, 2], result$stimuli[[2]]$w2, ylim = c(-2,
+ 2), pch = 20, cex = 1.5, cex.lab = 1.6, bty = "n",
+ xlab = "True W, second dimension", ylab = "Recovered W, second dimension")

```

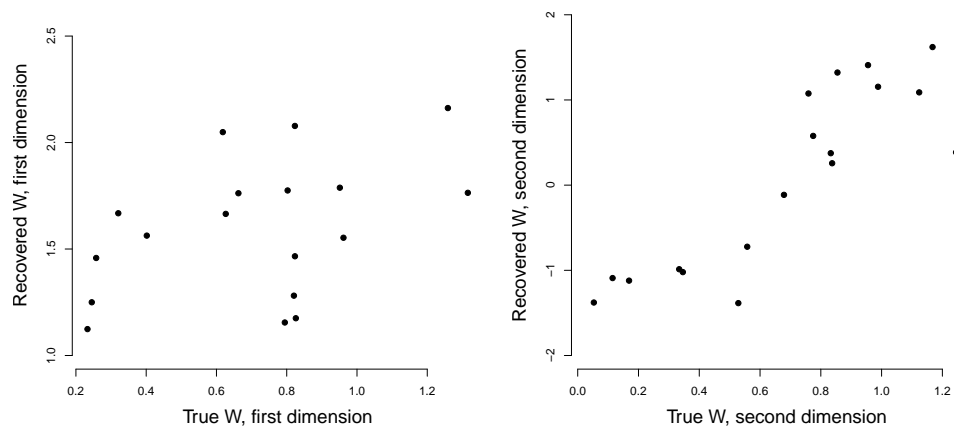


Figure 2: Plots of True vs. Estimated W scores, first and second dimension.

```
> par(mfrow = c(1, 1))
> plot(c, result$stimuli[[2]]$c, pch = 20, cex = 1.2,
+      cex.lab = 1.1, bty = "n", xlab = "True C",
+      ylab = "Recovered C")
```

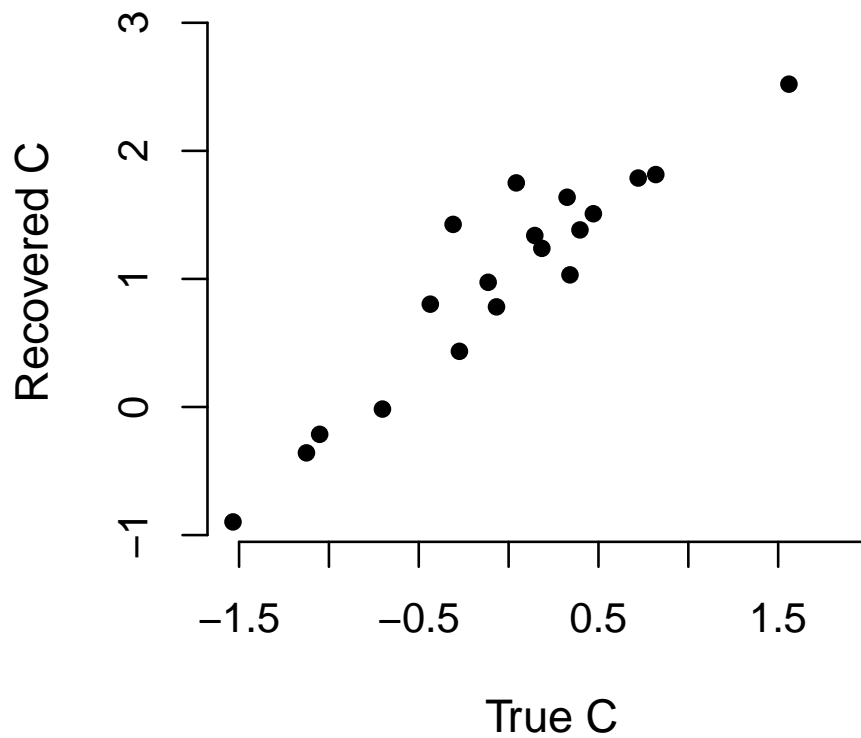


Figure 3: Plot of True vs. Estimated c scores.

```

> W.hat <- cbind(result$stimuli[[2]]$w1, result$stimuli[[2]]$w2)
> Psi.hat <- cbind(result$individuals[[2]]$c1, result$individuals[[2]]$c2)
> X.hat <- Psi.hat %*% t(W.hat) + Jn %o% result$stimuli[[2]]$c
> par(mfrow = c(1, 2))
> plot(X.true[missing], X.hat[missing], pch = 20,
+      cex = 0.4, cex.lab = 1.2, bty = "n", xlab = "True X, missing values",
+      ylab = "Recovered X, missing values")
> plot(X.true[!(1:(N * M) %in% missing)], X.hat[!(1:(N *
+      M) %in% missing)], pch = 20, cex = 0.4, cex.lab = 1.2,
+      bty = "n", xlab = "True X, nonmissing values",
+      ylab = "Recovered X, nonmissing values")

```

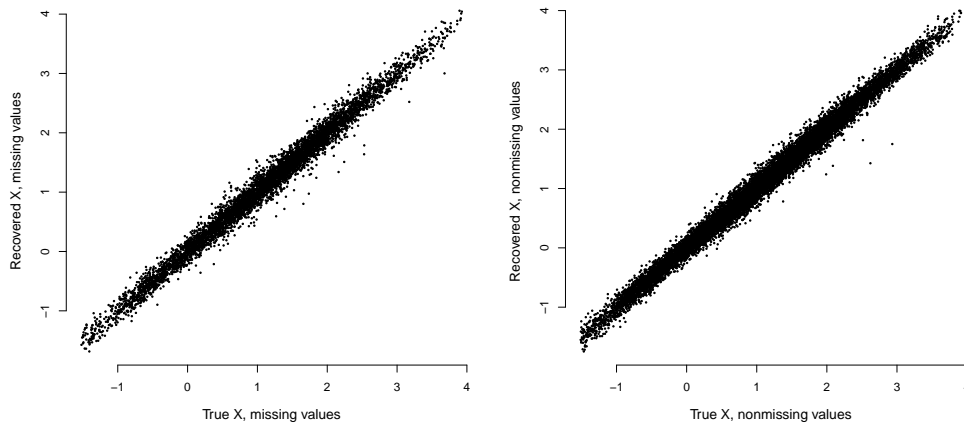


Figure 4: Plots of True vs. Estimated X scores for missing vs. nonmissing values.

similar in both plots). While further tests are necessary, these results suggest that the use of the techniques demonstrated here may have greater applicability beyond survey research. Further discussion of imputation can be found in the unpublished appendix to Poole (1998).

4 Recovering a Basic Space with the 1980 National Election Study Issue Scales

In this section we present an application of the basic space model to a common problem in social science. One issue of interest to political scientists in particular are empirical models of voting (also known as ideal point models), that allow legislator locations in an abstract policy or ideological space to be inferred from their roll call votes. The recovered scores have wide applicability to the study of Congress (Poole and Rosenthal 1997; McCarty et al. 2006), elections (Herron and Lewis 2007), courts (Martin and Quinn, 2002), and in non-

legislative voting bodies such as the United Nations (Voeten 2000).⁸ The most prominent ideal point model in the political science literature is NOMINATE (Poole and Rosenthal 1997), a model that estimates the policy preferences of legislators using observed roll call votes as the primary source of data.⁹ However, the use of models such as NOMINATE may not always be possible because roll call data is often not available or recorded. In such instances, the basic space model shown here presents an attractive alternative estimator.¹⁰

We present a simple example that applies the basic space model to a set of issue scales from the 1980 National Election Study. This survey contains $N=1614$ respondents who were asked to place themselves on scales about desired levels of defense spending, inflation, tax cuts, abortion, liberal-conservative scales, the role of women, the role of government in providing jobs, busing, and other similar issues. We assume that each respondent has a location in a common ideological space and attempt to recover estimates of those locations, which is represented as Ψ in (1). In providing responses to the issue scales, each respondent reports their true ideological position Ψ , modified by a stretch parameter W and an additive intercept parameter c along with noise E_0 . The data is simply stored in a standard matrix or data frame with respondents on the rows and survey questions (i.e. stimuli) on the columns as follows:

```
> data(Issues1980)
> Issues1980[1:10, 1:4]
```

	libcon1	defense	govserv	inflation
1	0	7	5	4
2	4	4	6	7
3	6	3	0	0
4	5	6	2	8
5	3	4	2	4
6	5	5	4	0
7	8	2	6	5
8	2	7	7	6
9	6	7	2	2
10	5	4	2	5

Virtually all surveys contain missing data, and for the two survey questions about abortion, ‘7’ is used as a missing data code. However, many of the other scales in this data set use 7 point scales, so we need to recode the missing data for those questions.

```
> Issues1980[Issues1980[, "abortion1"] == 7, "abortion1"] <- 8
> Issues1980[Issues1980[, "abortion2"] == 7, "abortion2"] <- 8
```

⁸For a more extensive review of applications of spatial modelling in the social sciences, see Poole (2005).

⁹The package *wnominate* on CRAN contains software used to estimate NOMINATE scores.

¹⁰For a more comprehensive review of the advantages and disadvantages of different data sources for spatial models, see Saiegh (2009).

Estimation of the scores is now trivial using the *blackbox()* function, which takes the same arguments already described in the Monte Carlo example:

```
> result <- blackbox(Issues1980, missing = c(0,
+      8, 9), verbose = FALSE, dims = 3, minscale = 8)
```

Objects of class *blackbox()* can also be summarized using the *summary()* function, although the summaries largely provide only summaries of the stimuli. For each dimension estimated, the summary provides the intercept (*c*) and stretch ($w_1 \dots w_3$) parameters for each question, as well as the number of respondents and various fit statistics.

```
> summary(result)
```

SUMMARY OF BLACKBOX OBJECT

```
-----
```

	N	c	w1	R2
libcon1	875	4.280	-3.028	0.414
defense	1163	5.210	-1.754	0.123
govserv	1119	4.323	4.302	0.450
inflation	816	4.106	2.015	0.159
abortion1	1238	2.856	0.627	0.031
taxcut	836	2.839	-1.074	0.055
libcon2	949	4.369	-2.755	0.414
govhelpmin	1160	4.542	-3.400	0.412
russia	1152	3.891	-3.034	0.231
womenrole	1223	2.845	-2.866	0.204
govjobs	1131	4.377	-4.488	0.518
equalrights	1144	2.663	-3.297	0.381
busing	1219	6.051	-2.699	0.255
abortion2	1246	2.675	0.724	0.047

	N	c	w1	w2	R2
libcon1	875	4.300	-2.966	0.954	0.424
defense	1163	5.214	-1.779	0.899	0.147
govserv	1119	4.368	4.331	3.042	0.617
inflation	816	4.152	2.088	2.940	0.393
abortion1	1238	2.856	0.512	-2.211	0.290
taxcut	836	2.818	-1.103	-0.667	0.071
libcon2	949	4.377	-2.758	0.459	0.423
govhelpmin	1160	4.535	-3.456	-0.119	0.424
russia	1152	3.887	-3.140	0.241	0.247
womenrole	1223	2.872	-2.466	6.007	0.771
govjobs	1131	4.350	-4.595	-2.417	0.635

equalrights	1144	2.673	-3.148	2.438	0.491	
busing	1219	6.049	-2.741	0.059	0.263	
abortion2	1246	2.676	0.629	-2.112	0.318	
	N	c	w1	w2	w3	R2
libcon1	875	4.294	-2.976	0.708	-1.180	0.448
defense	1163	5.200	-1.806	1.586	2.562	0.315
govserv	1119	4.410	4.295	3.707	2.929	0.778
inflation	816	4.169	1.998	3.286	1.111	0.451
abortion1	1238	2.856	0.497	-2.004	1.174	0.312
taxcut	836	2.813	-1.049	-0.902	-0.891	0.091
libcon2	949	4.367	-2.785	0.265	-0.557	0.437
govhelpmin	1160	4.534	-3.457	0.140	0.961	0.440
russia	1152	3.831	-3.255	1.558	5.590	0.695
womenrole	1223	2.891	-2.372	5.602	-2.868	0.805
govjobs	1131	4.341	-4.632	-2.176	1.392	0.648
equalrights	1144	2.680	-3.159	1.860	-2.372	0.563
busing	1219	6.042	-2.819	0.329	1.282	0.306
abortion2	1246	2.675	0.587	-1.980	0.906	0.329

```

Dimensions Estimated: 3
Number of Rows: 1270
Number of Columns: 14
Total Number of Data Entries: 15271
Number of Missing Entries: 2509
Percent Missing Data: 14.11%
Sum of Squares (Grand Mean): 52705.13

```

When using *blackbox()* for applied research, the researcher's principal goal is the recovery of the individual parameters stored as the *individuals* data frame. These typically represent our estimate of the individual's ideological measure in the basic space. Due to the issues with model identification discussed earlier, these measures are measured only up to an affine transformation of the true space. In particular, the rotation of the estimate is not specified, so if the ideological measure is to be substantively measured as a liberalism/conservatism score, its rotation should be validated so that it can be transformed if necessary. Here we conduct such a check by correlating our recovered scores with self-reported liberal-conservative scores, where higher scores indicate higher levels of conservatism. The correlation is negative, suggesting that as the recovered scores increase, the respondents become more liberal. Since the norm in political science research is to orient liberal-conservative scores to increase as conservatism increases, the researcher may wish to rotate the scores (i.e. by multiplying them by -1) before using them for auxiliary analyses.

```
> cor(result$individuals[[1]]$c1, Issues1980[, "libcon1"])
```

[1] -0.1959135

5 Recovering a Basic Space with the 1980 National Election Study Liberal-Conservative Scales

In our previous example applying the basic space model to analyze respondent self-placement on issues scales, we considered an example where the bias and stretch parameters c and w were estimated for the column parameters. However, we may instead wish to estimate a version of the model where c and w are estimated for the row parameters (i.e. the survey respondents) instead. This is simply a transposed version of the basic space model, where $m > n$ instead of $n > m$. A transposed model may be reasonable in cases where the survey data to be used is perceptual data of the stimuli of interest. In this example we analyze perceptual data from the 1980 National Election Study. A total of $N=888$ respondents were asked to place six stimuli (Carter, Reagan, Kennedy, Anderson, the Republicans, and the Democrats) on a 7 point liberal-conservative scale. Our objective is to estimate the locations of the six stimuli in the basic space, which each respondent perceives with some bias and stretch parameter. The data is input in a manner identical to before, with survey respondents on the rows and stimuli on the columns. One very important difference between `blackbox()` and `blackbox_transpose()` is that in most survey data sets, the number of respondents is very large relative to the number of stimuli. This typically means that `blackbox_transpose()` takes much longer to estimate because each respondent estimates both a bias c and stretch W parameter. To estimate the 1980 liberal-conservative placements using `blackbox_transpose()`, we simply load the data and call the estimator as follows:

```
> data(LC1980)
> LCdat = LC1980[, -1]
> LCdat[1:10, ]
```

	Carter	Reagan	Kennedy	Anderson	Republicans	Democrats
1	2	6	1	7	5	5
8	4	6	4	7	6	4
9	3	6	3	3	6	2
10	6	4	3	3	5	4
11	7	2	5	5	7	5
13	6	6	2	5	7	4
14	3	6	2	5	6	3
16	3	7	4	2	7	3
17	5	3	5	2	8	8
19	3	6	4	5	6	2

```
> result <- blackbox_transpose(LCdat, missing = c(0,
+ 8, 9), dims = 3, minscale = 5, verbose = TRUE)
```

```

> par(mfrow = c(1, 2))
> plot(result)
> plotcdf.blackbt(result)

```

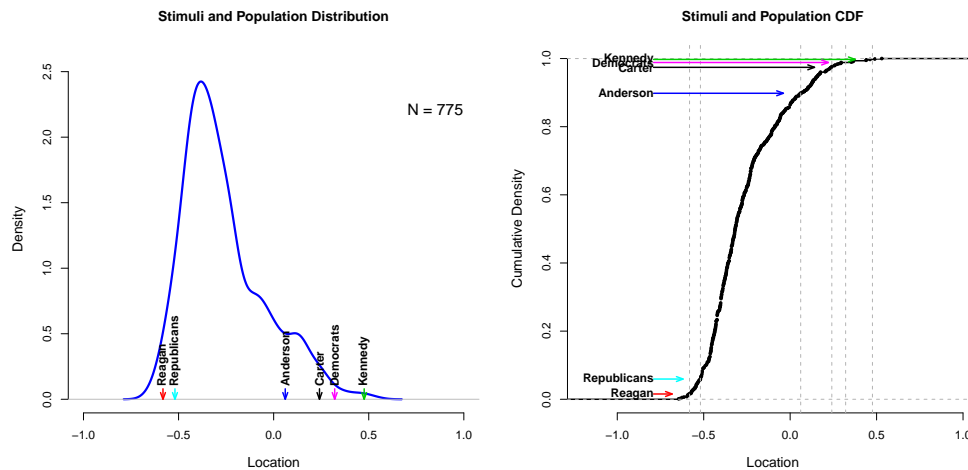


Figure 5: Blackbox Transpose PDF and CDF plots.

Beginning Blackbox Transpose Scaling...6 stimuli have been provided.

Blackbox-Transpose estimation completed successfully.

In an effort to simplify interpretation of results from *blackbox_transpose()*, we include two plot functions. These functions plot the location of the stimuli against a probability and cumulative distribution plot of locations of the population weights. These weights represent reasonable estimates of the two-dimensional ideal points, as shown in Abrajano and Poole (2010).

We can also produce summary reports of the stimuli as follows:

```

> summary(result)

```

SUMMARY OF BLACKBOX TRANSPOSE OBJECT

```

-----

```

	N	coord1D	R2
Carter	768	0.241	0.563
Reagan	765	-0.582	0.822
Kennedy	754	0.476	0.648
Anderson	689	0.061	0.230
Republicans	771	-0.519	0.757
Democrats	774	0.321	0.651

	N	coord1D	coord2D	R2
Carter	768	0.238	-0.407	0.720
Reagan	765	-0.580	-0.101	0.839
Kennedy	754	0.481	0.013	0.680
Anderson	689	0.059	0.864	0.946
Republicans	771	-0.518	-0.117	0.767
Democrats	774	0.321	-0.252	0.718

	N	coord1D	coord2D	coord3D	R2
Carter	768	0.191	-0.261	-0.663	0.918
Reagan	765	0.216	0.556	0.141	0.856
Kennedy	754	0.162	-0.510	0.697	0.981
Anderson	689	-0.911	0.053	-0.002	1.000
Republicans	771	0.210	0.498	0.055	0.780
Democrats	774	0.131	-0.335	-0.228	0.765

```

Dimensions Estimated: 3
Number of Rows: 6
Number of Columns: 775
Total Number of Data Entries: 4521
Number of Missing Entries: 129
Percent Missing Data: 2.77%
Sum of Squares (Grand Mean): 12683.93

```

The transposed basic space model is a generalization of the original model by Aldrich and McKelvey (1977) to matrices with missing values and estimates with more than one dimension. For historical purposes, we include the original Aldrich-McKelvey estimator with this package, and estimate the same model using it as shown below. While the *aldmck()* function accepts nearly identical arguments to the *blackbox_transpose()*, one notable difference appears by default. In addition to ranking of stimuli, *aldmck()* also accepts a column (specified by the argument *respondent*) in the data matrix that specifies the respondent's self placements on the issue scale. The reported respondent rating is then transformed into an ideology score by applying the respondent's personal stretch and bias parameters to that score.

```

> data(LC1980)
> result <- aldmck(data = LC1980, polarity = 2,
+   respondent = 1, missing = c(0, 8, 9), verbose = TRUE)

```

Beginning Aldrich-McKelvey Scaling...

```

Column 'Self' is set as the self placement.
Column 'Carter' is set as the left-leaning stimulus.

```

646 of 888 observations are complete.
6 stimuli have been provided.

Aldrich-McKelvey estimation completed successfully.

```
> summary(result)
```

```
SUMMARY OF ALDRICH-MCKELVEY OBJECT
```

```
-----  
Number of Stimuli: 6  
Number of Respondents Scaled: 643  
Number of Respondents (Positive Weights): 569  
Number of Respondents (Negative Weights): 74
```

```
R-Squared: 0.65  
Reduction of normalized variance of perceptions: 0.14
```

	Location
Kennedy	-0.485
Democrats	-0.317
Carter	-0.232
Anderson	-0.065
Republicans	0.517
Reagan	0.582

```
> plot.aldmck(result)
```

6 Conclusion

Social scientists often wish to infer the locations of voters and legislators in an abstract policy or ideological space. In situations where this cannot be accomplished with a preference data-oriented estimator such as NOMINATE (Poole and Rosenthal, 1997), the use of perceptual data-oriented estimators such as the basic space technique described here are a useful alternative. In fact, it is somewhat surprising that given the abundance of perceptual data questions found in most political science surveys, articles that use perceptual data are still uncommon in the social science literature. By providing an R (R Development Core Team, 2010) package that facilitates the analysis of perceptual data in a popular statistics environment, we hope to encourage a renewed interest in this important literature.

References

- [1] Abrajano, Marisa and Keith T. Poole (2010). “A Method of Linking Surveys Using Affective Signatures with an Application to Racial/Ethnic Groups in the U.S.” Working Paper.
- [2] Aldrich, John H. and Richard D. McKelvey (1977). “A Method of Scaling with Applications to the 1968 and 1972 Presidential Elections.” *American Political Science Review* 71(1), 111-130.
- [3] Carroll, Douglas and Jih-Jie Chang (1970). “Analysis of Individual Differences in Multidimensional Scaling Via an N-Way Generalization of “Eckart-Young” Decomposition.” *Psychometrika* 35:283-319.
- [4] Eckart, Carl and Gale Young (1936). “The Approximation of One Matrix By Another of Lower Rank.” *Psychometrika* 1, 211-218.
- [5] Herron, M. C. and J. B. Lewis (2007). “Did Ralph Nader Spoil Al Gore’s Presidential Bid? A Ballot-Level Study of Green and Reform Party Voters in the 2000 Presidential Election.” *Quarterly Journal of Political Science* 2, 205-226.
- [6] Martin, A. D. and K. Quinn (2002). “Dynamic Ideal Point Estimation via Markov Chain Monte Carlo for the U.S. Supreme Court, 1953-1999.” *Political Analysis* 10, 134-153.
- [7] McCarty, Nolan., Keith T. Poole, and Howard Rosenthal (2006). *Polarized America: The Dance of Ideology and Unequal Riches*. Cambridge: MIT Press.
- [8] Palfrey, Thomas R. and Keith T. Poole (1987). “The Relationship between Information, Ideology, and Voting Behavior.” *American Journal of Political Science* 31(3), 511-530.
- [9] Poole, Keith T. (1998). “Recovering a Basic Space From a Set of Issue Scales.” *American Journal of Political Science* 42(3), 954-993.
- [10] Poole, Keith T. (2005). *Spatial Models of Parliamentary Voting*. Cambridge: Cambridge University Press.
- [11] Poole, Keith T. and Howard Rosenthal (1997). *Congress: A Political-Economic History of Roll Call Voting*. New York: Oxford University Press.
- [12] R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing <http://www.R-project.org>.
- [13] Saiegh, Sebastian (2009). “Recovering a Basic Space from Elite Surveys: Evidence from Latin America.” *Legislative Studies Quarterly* 34 (1), 117-145.
- [14] Takane, Yoshio, Forrest Young, and Jan de Leeuw (1977). “Nonmetric Individual Differences Multidimensional Scaling: An Alternative Least Squares Method with Optimal Scaling Features.” *Psychometrika* 42, 7-67.

- [15] Voeten, Erik (2000). “Clashes in the assembly.” *International Organization* 54 (2), 185-215.