# **mhdiff**: User Documentation

David Strang[1]
Department of Sociology
Cornell University

August 1995
(last revised January 2002)

---

[1] Software development for **mhdiff** was supported by National Science Foundation grant SES-9213152. I thank Tim Dorcey for his programming advice and assistance.

**mhdiff**: User Documentation

**mhdiff** is a *SAS*-IML routine that estimates a class of **M**ultiplicative **H**eterogeneous **DIFF**usion models. The most recent non-experimental sendable version is *mhdiffb2* in *mhdiff_b.sas*.

Contact:

David Strang
Dept of Sociology
Uris Hall
Cornell University
Ithaca NY 14853
(607) 255-9533
email: ds20@cornell.edu

# 1 Model Definition

**mhdiff** constructs covariates for and estimates models of the form:

$$h_n(t) = \exp[\alpha X_n + \sum_{s \in \mathcal{S}_n(t)} (\beta V_n + \gamma W_s + \delta Z_{ns})] \tag{1}$$

where

$h_n(t)$ is the hazard of an event of interest for case $n$ at time $t$.

$X_n$ is a covariate vector describing the *intrinsic propensity* of $n$ to experience the event (to adopt), net of diffusion influences.

$\mathcal{S}_n(t)$ is the set of prior adopters who influence $n$.

$V_n$ is a covariate vector describing the *susceptibility* of $n$ to diffusive influences from $\mathcal{S}(t)$.

$W_n$ is a covariate vector describing the *infectiousness* of $s$ in influencing all $n$.

$Z_{ns}$ is a covariate vector describing the *social proximity* of $n$ and $s$ (the pairwise-specific influence of $s$ on $n$).

See Strang and Tuma "Spatial and Temporal Heterogeneity in Diffusion," **American Journal of Sociology 99** (1993): 614-39 for discussion of this class of models.

Note that this program can be used to construct variables, such as infectiousness or social proximity effects, to be incorporated in analyses utilizing other software (for example, if one wanted to perform a partial likelihood analysis or some other methodology not avaliable in *mhdiff*).

## 2 Running mhdiff

*mhdiff* is an uncompiled *SAS* macro (version 6.09 and above). It is designed to be run in conjunction with a calling *SAS* program that reads the data to be analyzed, defines global variables, and identifies the model to be estimated.

The program is written as a PROC IML routine. All data access and variable manipulation thus follow the rules of IML notation, and a knowledge of basic IML syntax is useful in specifying models. See the appendix below for a short overview of matrix definition in IML.

*mhdiff* calls a Newton-Raphson nonlinear optimizer provided by NLP (NLPNRA) to perform numerical estimation. NLPNRA takes a pure Newton step when the Hessian is positive definite and the step reduces the value of the objective function; otherwise a combination of ridging and line search are used. Analytic first and second derivatives are supplied to NLPNRA by *mhdiff*. For details on numerical estimation, see Wolfgang Hartmann, *Internal Draft Document: Nonlinear Optimization in IML Revised: Release 6.09* and *The NLP Procedure: Release 6.08 Extended User's Guide* (*SAS* Institute: Cary NC).

Prior to invocation, the calling program should define the values of relevant parameters via a series of global variable declarations. These take the form of statements like:
**% let** $x = y$**;**
where **x** is the name of a variable defined in mhdiff, and **y** is the value of that variable in a particular application.

The *SAS* calling program invokes *mhdiff* via the lines

**include mhdiff_b/nosource2;**
**mhdiffb2;**

This assumes *mhdiff_b.sas* is a file in the current directory; it should be amended if *mhdiff_b.sas* is located elsewhere.

## 2.1 External dataset access

*mhdiff* may operate with *SAS* datasets located in external files or with 'working' datasets defined by the calling program. All datasets utilised by *mhdiff* must be in one type of location or the other. All external *SAS* datasets must be located in the same directory.

Dataset definitions include:

**indir**

- 1 = data is located in external file(s)
- 0 = data is located in working dataset(s)

**difflib** name of directory where external datasets are located.

**sdata** name of spell-organized dataset holding the event history and individual-level covariates.

There are also two datasets with fixed names (not defined by the user); one for data written by Mhdiff and one for proximity matrix data (not all applications will utilize this data format, but thise that do will need to label the datasets in the way expected by Mhdiff).

**fromdiff** *SAS* dataset constructed by *mhdiff* out of the start and end times, end state, case id, and explanatory covariates used to estimate the specified hazard model.

**nxnd_k** name(s) of datasets holding case-by-case proximity matrices. 'k' identifies which of the modeled social proximity effects the case-by-case matrix refers to. For example, if data for the 3rd social proximity effect in the model is held in a proximity matrix, then the name of the *SAS* dataset holding that proximity matrix must be **nxnd_3**.

# 3 Global Variables

## 3.1 Spell Identifiers

In general, covariate definitions index into the spell-organized dataset **sdata**. They do so on the basis of the order of the variables in that *SAS* dataset. Note, however, that only numeric variables are retained in **sdata** when it is read by *mhdiff*, since character and string variables are always dropped when a *SAS* dataset is read into IML as a numerically-valued matrix.[1]

The location within **sdata** of the following *time* and *state* variables must be defined to allow *mhdiff* to interpret the event history format of **sdata**.

**st** start time (time spell begins)

**et** end time (time spell ends)

**ss** start state (value of outcome of interest at st)

**es** end state (value of outcome of interest at et)

**id** case id (numeric case identifier)

**sort_var** name of the variable that *mhdiff* will sort the data by.
If **use_et2** = 0 this is the SAS variable name of **et**;
if **use_et2** = 1 this is the SAS variable name of **et2**.

Spells are censored if the value of **ss** equals the value of **es**.

*mhdiff* is limited to examining one transition in each analysis. It will treat all end state values different from that of the start state as equivalent events, and will not distinguish between start states. Examination of multiple transitions can proceed in a competing risk framework requiring $n$ distinct analyses for $n$ transitions, where in each analysis all cases experiencing transitions other than the one under investigation are treated as censored.

---

[1] For example, if a *SAS* dataset consisted of three variables: an alphanumeric case identifier, a numeric variable X, and a numeric variable Y, then for the purposes of IML (and *mhdiff*) the dataset would be converted into a 2 column numeric matrix, with column 1 holding X and column 2 holding Y.

## 3.2 Diffusion options

The following variables specify the set of relevant influencers $\mathcal{S}_n(t)$:

**sametime** (scalar) = 0 if simultaneous influence disallowed, = 1 if allowed

(Simultaneous influence implies that events that occur at the same time affect each other)

**simlapse** (scalar) Postulated lag in influence of simultaneous events. Only meaningful if sametime equals 1. Must be set to a value not greater than the smallest spell length in the data.

**insum** (scalar) = 0 if $\mathcal{S}_n(t)$ includes all prior (or simultaneous, if sametime = 1) adopters; = 1 if $\mathcal{S}_n(t)$ varies over $n$.

**igroup** (vector) gives location of variables whose values for $s$ must equal those for $n$ if $s$ is to be treated as influencing $n$ (i.e., if $s$ is included in $S_n(t)$).

Additional diffusion options include:

**mvalue** (scalar) user-defined missing value for case or group ids.

**tiesize** (matrix) gives location of variables that indicate the magnitude of the tie between $i$ and $j$. Used when want to weight influences drawn from a linked list. Dimension is **dterm** by the number of possible ties (length of the longest list).

Secondary event types:

**use_et2** (scalar) = 1 if want to compute diffusion covariates based on a second event type (not the same event that is being modeled);
= 0 if not

**ss2** (scalar) index of the start state of the second event type

**es2** (scalar) index of the end state of the second event type

**st2** (scalar) index of the start time of the second event type

**et2** (scalar) index of the end time of the second event type

### 3.3 System options

**wspace** the amount of workspace allocated by IML

**verbose** print supplementary descriptive statistics and other information

### 3.4 Estimation options

*mhdiff* calculates initial estimates for parameters unless these are supplied by the user (see below). The initial estimate for $\alpha_0$ is calculated as the number of events in the sample divided by total time at risk in the sample (spell lengths summed across all spells for all cases). Other parameters are set to 0.

**user_est** (scalar) = 1 if user will supply initial estimates (starting values for parameters);
= 0 if starting values should be calculated by *mhdiff*.

**init_est** (vector) of initial estimates provided for model parameters. Only specified if user_est = 1. The vector has the same length as the number of parameters to be estimated.

## 4 Model Specification

Each of the 4 parameter vectors defined above (propensity, susceptibility, infectiousness, social proximity) is specified through a scalar global variable indicating the existence or type of the parameter vector, and a vector-valued global variable indicating the location of the covariates $(X_n, Y_n, W_s, Z_{ns})$ which define the components of the parameter vector. Parameters for social proximity effects require additional specification.

*Propensity setup*: $\alpha X_n$

**aterm** scalar giving existence of the propensity vector

- 0 : no propensity vector in model
- 1 : propensity intercept, no covariates in model
- 2 : propensity covariates, no propensity intercept in model

- 3 : both propensity intercept and covariates in model.

**avars** indices of propensity covariates (locations in dataset).

*Susceptibility setup*: $\beta Y_n$

**bterm** scalar giving type of the susceptibility vector

- 0 : no susceptibility term
- 1 : susceptibility intercept, no susceptibility covariates
- 2 : susceptibility covariates, no susceptibility intercept
- 3 : both intercept and susceptibility covariates.

**bvars** indices of susceptibility covariates (locations in dataset).

*Infectiousness setup*: $\gamma W_s$

**cterm** scalar giving type of the infectiousness vector

- 0 : no infectiousness term
- 1 : infectiousness covariates. No intercept is implied or permitted.

**cvars** indices of infectiousness covariates (locations in dataset).

*social proximity setup*: $\delta Z_{ns}$

**dterm** scalar giving *number* of the social proximity *effects* (equals the number of estimated parameters)

- 0 : no proximity terms
- k : k social proximity effects will be estimated. No intercept is implied or permitted.

**dopts** Gives the *type* of each social proximity effect, used to compute social proximity scores ($Z_{ns}$) from the variables indexed in **dvecs**. See below for specification.

dimension is **dterm** x 5.

**dvecs** a matrix whose $r$th row gives the indices of covariates used to compute the $r$th social proximity score.

dimension is **dterm** x (length of the longest d vector)

**dnames** gives names to each proximity effect to be estimated.

dimension is **dterm** x 1

## 4.1 Types of Social Proximity Effects

While the specification of covariates for propensity, susceptibility, and infectiousness effects is straight-forward, social proximity effects are more complicated. There are many ways to define and implement connections between prior adopter and potential adopter; the strategy here is to define alternative *measurement types* that permit effects to be either calculated from corresponding data values, to be drawn from a separate file that holds the value of all pairwise connections in a data matrix, or to be based on a linked list.

Four measurement types are defined. Two, the difference and absolute power metrics, are functions of two sets of covariate values: those of the influencee $i$ and the influencer $j$. The difference metric permits representation of asymmetric influence relations, where $j$ may influence $i$ but not vice versa. The absolute power metric permits representation of symmetric proximities, where the proximity of $i$ to $j$ equals the proximity of $j$ to $i$.

The other two measurement types (a full ($N$ $x$ $N$) influence matrix and a linked list) directly define influence relations rather than specifying a way of calculating them from covariate data. A full influence matrix provides the magnitude of influence for each ordered dyad. A list of linked cases identifies those cases $j$ which influence each $i$.

Note that the more flexible the measurement type, the more complicated data setup tends to be. Computing proximity based on corresponding data elements is inexpensive and advantageous whenever there is a simple rule that indicates how diffusion is hypothesized to work (for example, within homogeneous groups based on equivalent values for race or gender; or where actors of one type affect actors of another type but not vice-versa; or where geographical distance can be computed from codes locating each actor, as in a city-by-city distance table).

Where there is no clear rule that links $i$ and $j$ but relatively few $j$s affect each $i$, it is convenient to list influence relations via a linked list (for example, if each individual has been asked to name 3 friends, or if each firm has director interlocks with a modestly sized set of other firms). It is possible here to indicate the strength of each of the ties on such a list, rather than assume they all have the same strength.

If the pattern of hypothesized connections neither follows a rule nor is limited to a sparse set of connections (for example, if everyone potentially affects everyone else), it is necessary to write out the strength of the influences in a full matrix whose size equals the number of actors.

## 4.2 Measurement Types

### 4.2.1 Difference Metric

A difference metric sets up a non-linear difference relation between $k$ variables $x_{ik}$, $x_{jk}$, where $i$ indexes the influencee (potential adopter) and $j$ the influencer (prior adopter), as follows:

Two parameters, $p$ and $h$ define the type of difference metric requested.

- if $p \neq 0$ and $h = .$ (*SAS* missing value)

  $z_{ij} = \sum_{k=1,K} (x_{jk} - x_{ik})^p$

- if $p \neq 0$ and $h \neq .$

  $z^*_{ij} = \sum_{k=1,K} (x_{jk} - x_{ik})^p$

  $$\text{where} \quad \begin{array}{ll} z_{ij} = z^*_{ij} & \text{if } z^*_{ij} > h \\ z_{ij} = 0 & \text{if } z^*_{ij} \leq h \end{array}$$

- if $p = 0$ and $h = .$

  $z^*_{ij} = \sum_{k=1,K} (x_{ik} - x_{jk})$

  $$\text{where} \quad \begin{array}{ll} z_{ij} = 1 & \text{if } z^*_{ij} > 0 \\ z_{ij} = 0 & \text{if } z^*_{ij} = 0 \\ z_{ij} = -1 & \text{if } z^*_{ij} < 0 \end{array}$$

- if $p = 0$ and $h \neq .$

  $z^*_{ij} = \sum_{k=1,K} (x_{ik} - x_{jk})$

  $$\text{where} \quad \begin{array}{ll} z_{ij} = 1 & \text{if } z^*_{ij} > h \\ z_{ij} = 0 & \text{if } z^*_{ij} \leq h \end{array}$$

### 4.2.2 Absolute Power Metric

An absolute power metric sets up a non-linear combination of $K$ variables $x_{ik}$, $x_{jk}$ where $i$ indexes the influencee and $j$ the influencer.

Two parameters, here denoted $p$ and $r$, again define the type of absolute power metric (for example, Euclidean or Minkowski).

Note that when $p \neq 0$ the absolute power metric yields a distance score (which can be converted to a proximity via dopts[,2] = 2, 3, or 4.)

- if $p \neq 0$ and $r \neq 0$

  $z_{ij} = (\sum_{k=1,K} \mid x_{ik} - x_{jk} \mid^{p})^{1/r}$

  Note that for

  | | |
  |---|---|
  | $p = r = 2$ | : Euclidean distance |
  | $p = 2, r = 1$ | : Squared Euclidean distance |
  | $p = r = 1$ | : City-block (Manhattan) distance |
  | $p = r$ | : Minkowski distance |

- if $p \neq 0$ and $r = 0$

  (Chebychev or maximum distance across $k$ elements.)
  $z_{ij} = \max_{k=1,K}(\mid x_{ik} - x_{jk} \mid^{p})$

- if $p = 0$ and $r \neq 0$

  (Number of equivalent elements.)
  $z_{ij} = \sum_{k=1,K} \mathrm{EQ}(x_{ik}, x_{jk})$

- if $p = 0$ and $r = 0$

  (Any non-equivalent element = 1, else 0.)
  $z_{ij} = \max_{k=1,K} \mathrm{EQ}(x_{ik}, x_{jk})$

### 4.2.3 Full Influence Matrix

An influence matrix provides the value of each influence relation in the form of a full matrix relating all case ids. An influence matrix is stored as a *SAS* dataset with $I + 1$ variables and $I$ cases, where $I$ is the number of unique case ids in the event history dataset **sdata**. Such a dataset must be defined with a name of the form *nxnd_k*, where $k$ indexes the social proximity effect (the row in **dopts**) for which the influence matrix provides proximities.

The first variable in *nxnd_k* identifies the case id structure of the influence matrix, such that the value of its $m$th entry gives the case id referred to in the $m$th row and column of the influence matrix. The influence matrix thus does not need to be sorted by **id** or in the same order as **sdata**.[2] The remaining I columns of *nxnd_k* form a square influence matrix whose $(i, j)$ entry represents the closeness of $i$ and $j$ (either the influence of $i$ on $j$ or the influence of $j$ on $i$, depending on the value of **dopts**[**,2**].

---

[2] Case ids may appear in *nxnd_k* that do not appear in **sdata**, though these will not have any effect on estimation. The converse is not true: a program warning will result if a case id found in **sdata** does not appear in the first column of *nxnd_k*.

## 4.3 Specification of Social Proximity Effects

The $r$th row of **dopts** indicates how to compose the $r$th social proximity score (from the elements of the $r$th row of **dvecs**). **dopts** has 5 components (columns):

**dopts column 1** measure type

- 0 = difference metric;
- 1 = absolute power metric;
- 2 = full (N x N) influence matrix;
- 3 = linked list.

**dopts column 2** direction of effect, given computed value $z_{ij}$

1. If measure type = 0
    - $1 : p_{ij} = z_{ij}$
2. If measure type = 1
    - $1 : p_{ij} = z_{ij}$
    - $2 : p_{ij} = p^*_{ij} = \begin{array}{ll} = 1 & \text{if } z_{ij} = 0 \\ = 0 & \text{if } z_{ij} \neq 0 \end{array}$
    - $3 : p_i = (\sum_{j \in \mathcal{S}_i(t)} p^*_{ij}) / \sum \mathcal{S}_i(t)$
      where the numerator is the number of potential influencers with a distance of zero from $i$ and the dividend is the size of the set $\mathcal{S}_i(t)$.
    - $4 : p_{ij} = \mathrm{Max}_{z_{ij}} - z_{ij}$
      where $\mathrm{Max}_{z_{ij}}$ is the largest $z_{ij}$ across all $i$ and all $j$.
3. If measure type = 2
    - $1 :$ $z_{ij}$ gives influence of adoption by $j$ on the hazard for $i$ (column on row)
    - $2 :$ $z_{ij}$ gives influence of adoption by $i$ on the hazard for $j$ (row on column)
4. If measure type = 3
    - $1 :$ **sdata[i,dvars]** identifies the cases that $i$ influences

- 2 : **sdata[i,dvars]** identifies the cases that $i$ is influenced by.

**dopts column 3**

1. If measure type = 0:
   - $p$ : $p$th power in difference metric
2. If measure type = 1:
   - $p$ : $p$th power in absolute power metric
3. If measure type = 2:
   - 1 : $p_{ij} = \text{Max}_{z_{ij}} - z_{ij}$ // (deviate scores from the maximum score in the matrix).
   - 2 : $p_{ij} = \text{Max}_{z_i} - z_{ij}$ // (deviate scores for each $i$ from the largest score for $i$).
   - 3 : $p_{ij} = z_{ij} / \sum_{k=1,K}^{k \neq i} z_{ik}$ // (standardize scores by dividing by sum of all scores for $i$)
   - 4 : $p_{ij} = (\text{Max}_{z_i} - z_{ij}) / \sum_{k=1,K}^{k \neq i} z_{ik}$ // (deviate and standardize by sum of original scores).
   - 5 : $p_{ij} = (\text{Max}_{z_i} - z_{ij}) / \sum_{k=1,K}^{k \neq i}((\text{Max}_{z_i} - z_{ik})$ // (deviate and standardize by sum of deviates).
4. If measure type = 3:
   - $g$ : index of variable in **sdata** that values in **dvars** refer to. If **dvars** is constructed as a list of case ids, then $g$ gives the index of the case id. If **dvars** is constructed as a list of groups, all members of which are proximate, then $g$ gives the location of the ‘group id’ variable in **sdata**.

**dopts column 4**

1. If measure type = 0:
   - h : threshold level in difference metric
2. If measure type = 1:
   - $r$ : $r$th root of absolute power metric
3. If measure type = 3:

- 0 : Social proximity sums the number of cases in *i*'s list who have adopted by *t*.
- 1 : Social proximity equals the number of cases in *i*'s list who have adopted by *t* divided by the size (length) of *i*'s list.
- 2 : Each tie located in **dvars** has a magnitude, given by the corresponding element in the matrix **tiesize**. Social proximity sums these magnitudes for cases in the list who have adopted by *t*.

**dopts column 5** Integer value giving the number of components of the proximity vector (specified in dvecs) that will be examined.[3]

Appendix: PROC IML Conventions

The matrix definition

A = {3 5, 2 4, 0 0 };

creates the matrix A equalling

$$\begin{pmatrix} 3 & 5 \\ 2 & 4 \\ 0 & 0 \end{pmatrix}$$

B = { 2 5 0 1 } thus defines a row vector; C = { 2, 5, 0, 1 } defines the corresponding column vector.

N = {"name 1" "name 2" } defines a 2 element character row vector.

A[1,2] equals the entry of A in the 1st row, 2nd column (valued as 5 above).

A[1,] equals the first row in A (valued as (3 5) above).

---

[3] Because **dvecs** must be a well-formed matrix, all its rows have the same number of columns, even when this is unnecessary to properly specify proximity scores. (For example, in a given model computation of the first proximity effect might require manipulation of 3 covariates, computation of the 2nd proximity effect might require manipulation of 1 covariate, and computation of the third proximity effect might require manipulation of 2 covariates.) To adjust for this, all operations in constructing the *r*th proximity score are conducted over the first k elements in the *r*th row of **dvecs**, where k = **dopts[r,5]**. Elements in the *r*th row of **dvecs** in columns k+1, k+2, etc are ignored.