APPENDIX AVAILABLE ON REQUEST

Research Report 140

Extended Follow-Up and Spatial Analysis of the American Cancer Society Study Linking Particulate Air Pollution and Mortality

Daniel Krewski et. al.

Appendix B. Algorithmic Description of the Cox Poisson Program

Note: Appendices Available on the Web appear in a different order than in the original Investigators' Report. HEI has not changed these documents.

Correspondence may be addressed to Dr. Daniel Krewski, McLaughlin Centre for Population Health Risk Assessment, Room 320, University of Ottawa, One Stewart Street, Ottawa, ON K1N 6N5, Canada. E-mail: cphra@uottawa.ca.

Although this document was produced with partial funding by the United States Environmental Protection Agency under Assistance Award CR–83234701 to the Health Effects Institute, it has not been subjected to the Agency's peer and administrative review and therefore may not necessarily reflect the views of the Agency, and no official endorsement by it should be inferred. The contents of this document also have not been reviewed by private party institutions, including those that support the Health Effects Institute; therefore, it may not reflect the views or policies of these parties, and no endorsement by them should be inferred.

This document was reviewed by the HEI Health Review Committee but did not undergo the HEI scientific editing and production process.

APPENDIX B: Algorithmic Description of the Cox-Poisson Program

Algorithmic Description of the Cox-Poisson Program

Edward Hughes

Introduction

The Cox-Poisson program is designed to carry out estimation of Cox Regression survival models, from data giving, among other things, the time to a certain event (e.g. death) for each subject in the study. It differs from the survival modules of general statistical systems such as SAS, S-Plus, R, and Stata in two main ways:

- It is designed to handle large data sets
- Random effects are allowed to have a more complicated covariance structure than most other programs support.

There are two ways to use the program: as a stand-alone system, and through an S interface, where we use "S" as a general term for the S language as implemented in the systems S-plus and R. The stand-alone version is somewhat restricted in what it allows, but it does all the important computational work, and this section will concentrate on it. We describe the structure of the data accepted by the program, the types of survival problem it can handle, and the estimation methods used and their computational implementation.

The statistical theory on which this program is based is derived from (Ma et al 2000). Since that paper uses somewhat different notation from what appears here, we give in the section "Remark on Notation" a brief guide to the notational differences.

Data Structure

Matrix and Vector Notation

If w is a column vector, we denote its i^{th} entry by a superscript, w^i. If x is a row vector, we denote its entries by subscripts: the j^{th} entry is x_j. If A is a matrix, we denote its i^{th} row by A^i, its j^{th} column by A_j, and the entry in the i^{th} row and j^{th} column by A^i_j.

Layout

The form of data accepted and its interpretation is very similar to the form accepted by the S survival code authored by T. Therneau and described in Therneau and Grambsch (2000). Suppose the time-dependent data is a list of records (data matrix M) of the following form:

V_1	$\cdots$	V_z	t_{start}	t_{end}	t_{org}	χ	R_1	$\cdots$	R_p

(1)

Here we have a start and end time for this particular record (we use "row" or "record" interchangeably), a status indicator χ, and values of the time-dependent covariates $R_1,\ldots,R_p$, considered constant on the time interval $(t_{start}, t_{end}]$. The columns $R_1,\ldots,R_p$ are considered a submatrix **R** of **M**, and its row k is denoted by $\mathbf{R}^k$. We have written the variables making up the data matrix in a conceptually convenient order, but in fact they can be in any order. The variables $V_1,\ldots,V_z$ consist of all other variables in the data set: they play various roles, for example defining strata, clusters, and individuals. We will generally assume that the data consist of records pertaining to individuals, with each individual represented by one or more records and identified by a value of one of the "V"

variables, which we will usually denote by e, but in fact this is not strictly necessary. If no individuals are identified in the data, then we can take $e^k = k$, i.e. identify individuals with records. With this understanding, we will refer to "individuals" without further qualification.

The status indicator χ codes the situation at the end of the interval, t_{end}: say, 0 for censored or incomplete (i.e. "no event"), and non-0 for failure (or "event") at t_{end}. We assume that the records for one individual specify disjoint time intervals. We denote by τ the "elapsed time" $t - t_{org}$, as will be explained later; similarly the interval

$$(\tau^k_{start}, \tau^k_{end}] = (t_{start} - t_{org}, t_{end} - t_{org}]$$

The reasons why an individual might have more than one record are:

1) Time-dependent covariates: These are represented as piecewise-constant in time. Each value, and the time interval over which it holds, will generate a separate record for an individual. The breakpoints are specific to an individual and to a variable; they can be different for different individuals. However, the fewer total breakpoints, the greater the computational efficiency.
2) Multiple events per individual: If an event can occur more than once (e.g. recurrent spells of a disease, or of unemployment), then the periods at risk and the events can be represented by multiple records per individual. We assume that, for each record (i.e. row) *k* of **M**, the corresponding individual is at risk of an event in the interval $(\tau^k_{start}, \tau^k_{end}]$, and the event occurs at τ^k_{end}, if at all in this interval. If the individual continues at risk after the event, a new record must be used to start that new at-risk spell. Whether the event occurs at τ^k_{end} or not is indicated by the value of χ^k.

3) Multiple periods at risk: If there are time intervals when the individual is not at risk (or not under observation), these intervals will be omitted from the data. The complementary time intervals, when the individual is at risk, will each generate one or more records for the individual. See the definition of the function δ below.

In the simplest case, with (at most) one event per individual, with the event removing the individual from further risk, and no time-dependent covariates, each individual would have only one record in the data set, which would cover the full interval of risk for this individual, ending with either an event or censoring, according to the value of χ.

The value t_{org} represents a time origin, which will usually be either 0 or the earliest t_{start} for the individual, but may also represent other things: in a multi-state model it will be the time of entering the current state. Most of the interest is in the elapsed times $\tau_t = t - t_{org}$: what we have been calling the failure times τ_{sh} are really these elapsed times, so that the earliest spell at risk usually starts at $\tau = 0$. As mentioned before, we will write τ_{end} for $t_{end} - t_{org}$.

Let $N_{\mathbf{M}}$ denote the number of rows of **M**.

Individuals at Risk

Let e^k denote the individual referred to in $\mathbf{M}^k$, i.e. row *k* of **M**. Let $\mathtt{E}$ denote the set of all individuals. Let us define a conceptual variable $\delta = \delta[e,t]$, or sometimes $\delta_e(t)$ which, for each individual e, is 1 if e is "at risk" and under observation at time t, otherwise 0. We assume that $\delta_e(t) = 1$ for *t* in the union of all the time intervals $(t_{start}, t_{end}]$ for e, and 0 for all other times. The data cover precisely the time intervals that

$\delta = 1$. This can be an arbitrary union of intervals. The use of δ is here just a notational convenience that makes the likelihood formulas simpler to state, but it plays an important role in the counting-process formulation of survival models. The usual notation for δ in the counting-process literature is $Y_e(t)$, but we use Y below for another purpose.

We also define the counting process $N[e,t]$, or sometimes $N_e(t)$, which counts the number of events observed for individual e in the interval $(t_{org}, t]$, i.e. in $(0, \tau]$. The events in question are those for which the individual is "at risk" in the sense of $\delta = 1$. We emphasize that *N* counts *observed* events, and δ indicates "under observation and at risk" for an *observable* event. If the individual is not under observation, then $\delta = 0$ and *N* remains constant, no matter what unobserved events befall the individual. We assume that δ and all the covariates R_j are continuous from the left in time, and that *N* is continuous from the right in time.

Time Origin

The time-origin variable is similar to the same variable in the Therneau code; t_{org} is a function of e and *t* also, i.e. $t_{org} = t_{org}[e,t]$, and we assume that

$$\delta[e,t] = 0 \text{ for } t \le t_{org}[e,t]$$

We denote by τ the "elapsed time", $\tau = t - t_{org}$, with *t* referring to "calendar time", and denote the elapsed-time indexes by τ:

$$\tau_{start} = t_{start} - t_{org}$$
$$\tau_{end} = t_{end} - t_{org}$$

Ordinarily we expect that $t_{org}[e,t]$ has jumps in *t* that occur infrequently, and is constant in *t* between jumps. It represents the beginning of the current spell of observed activity for the individual, and may be updated at every new spell, or it may not be. The issue here is the time interval over which the baseline hazard is parameterized. Consider a process in which an individual has recurrent spells under observation, interspersed with spells off observation. A spell may end with either an event or censoring. If each individual had only one such spell, we would ordinarily prepare the data placing the time origin t_{org} at the beginning of the spell, so each person's spell starts at $\tau = 0$. But with multiple spells per person, we may or not want to make this transformation. If we do make it, then we have a different value of t_{org} for each new spell, so that all of an individual's spells start at $\tau = 0$, and the baseline hazard would be defined on the τ-interval from 0 to the maximum spell length: the spells are in effect superimposed, and this normally involves an implicit assumption that different spells for the same individual are independent.

In some problems, on the other hand, we might not want to make that assumption, i.e. we might not want to redefine t_{org} at each new spell, but instead to consider the whole series of spells for each individual, leaving t_{org} fixed (i.e. varying only with individual). In this case, the baseline hazard is defined on a time interval that contains all the separate spells, not superimposed.

Strata

We allow time-dependent strata, which means in effect that a stratum *s* is not necessarily a set of individuals $\{e\}$, but a set of individual-time pairs $\{[e,t]\}$. Let $\phi^s(e,t)$ be the indicator of stratum *s*, i.e.

$$\phi^s(e,t) = \begin{cases} 1 & \text{if } [s,t] \in s \\ 0 & \text{if } [s,t] \notin s \end{cases}$$

We can state this more briefly by saying that a stratification is a partitioning of the set of rows of **M** into disjoint subsets, and each one of these subsets is a stratum. We will write $k \in s$ to mean that row *k* is in stratum *s*. Multi-state models can be implemented by the use of time-dependent strata: see Therneau and Grambsch (2000).

Risk and Event Sets

Form (once and for all) the list $F = \{\tau_1, \tau_2, \tau_3, \ldots, \tau_q\}$ of sorted distinct event-times (i.e. times for which $\chi = 1$). Note the distinction: the values $\{\tau_{end}^k\}$ are all the τ_{end} times, whether event or censoring; the values $\{\tau_h\}$ are just the event-times. If there is stratification, then we do this by strata: for each stratum *s*, we have a separate list $\{\tau_{s1}, \tau_{s2}, \tau_{s3}, \ldots, \tau_{sq_s}\}$ of distinct event-times in stratum *s*.

We define the risk set $R(\tau)$ in terms of the data matrix **M**:

$$R(\tau) = \{k \,|\, \tau \in (\tau_{start}^k, \tau_{end}^k]\}$$

where τ_{start}^k and τ_{end}^k are the τ-values (i.e. $t_{start} - t_{org}$ and $t_{end} - t_{org}$) for row *k* of **M**. In general, we indicate row *k* by a superscript. If there are strata, we can also define

$$R(\tau, s) = \{k \mid \tau \in (\tau^k_{start}, \tau^k_{end}] \ \& \ k \in s\}$$

Similarly, define the event-sets $D(\tau)$ as

$$D(\tau) = \{k \mid \tau = \tau^k_{end} \ \& \ \chi^k = 1\}$$
$$D(\tau, s) = \{k \mid \tau = \tau^k_{end} \ \& \ \chi^k = 1 \ \& \ k \in s\}$$

The event multiplicity $m(\tau)$ is the size of this *D*:

$$m(\tau) = \#(D(\tau))$$
$$m(\tau, s) = \#(D(\tau, s))$$

where $\#(S)$ means the number of members of the set *S*. We use the notation $R_{sh} = R(\tau_h, s)$, and similarly R_h, D_{sh}, D_h, m_{sh}, m_h. Note that both *R* and *D* are sets of rows of **M**. This is more general than defining them as sets of individuals, since each row corresponds to an individual and a time interval. Even though the sets *R* and *D* are sets of rows of **M** (i.e. sets of *k*), we shall sometimes abuse notation by writing $e \in R_{sh}$, where e is an individual.

Clusters

Suppose we have a nested system (tree) of clusters (sets of individuals, or of rows of **M**), ordered by the "$\subseteq$" relation. Let the "roots" or "level-1" clusters be those clusters with no parent (no larger cluster of which the given one is a subset). Let the "leaves", or "finest-level" clusters be those with no children. These are disjoint. A cluster can be both root and leaf. We define the level of a cluster λ recursively by:

- The level of a level-1 cluster is 1
- If the parent of λ has level ℓ, then λ has level $\ell + 1$.

The clusters of any level ℓ are disjoint. Let N_{leaf} be the total number of leaves. Ordinarily a 1-level clustering is defined by one variable V, say, with each value of V corresponding to one cluster. A two-level clustering is defined by two variables V_1 and V_2, say, so that the values of V_1 define the level-1 clusters, and within a level-1 cluster, the values of V_2 define the level-2 clusters. A L-level cluster tree is defined similarly by L variables. We can order leaf clusters lexically, by values of the variables defining them. Note that not all leaf clusters are necessarily at level L.

By a "leaf-vector" we mean a vector of dimension N_{leaf} whose components are associated with the corresponding leaf-clusters. For each level λ of the tree, there is a vector of random effects U_λ, but only the leaf-level random effects play a role in the estimation process: the lower levels are computed as part of post-processing, when estimation is complete. The notation U, with no level specified, will usually mean the leaf-level random effects.

For each row k of **M**, we denote by r^k the leaf cluster of which e^k is a member.

Primary and Secondary Data Tables

As mentioned above, time-dependent (TD) covariates, which are always considered piecewise-constant, can be given in the data by multiple records per individual, each record giving the value of the covariates on one time interval, with the non-time-dependent variables simply duplicated. For problems in which most of the covariates are time-dependent, this is usually the best way to organize the data. But if only one or two covariates out of, say, 30 or 40 are time-dependent, this is wasteful, since most of the information on each record will be redundant. In this case memory and (sometimes) run-

time can be saved by splitting the data into two tables, called “primary” and “secondary”. The primary table holds the non-time-dependent (NTD) covariates, along with the event indicators and other variables as described above for the matrix **M**. The secondary table lists the values of the TD covariates and corresponding time-intervals. It must also have a “key” variable, which determines the correspondence between the primary and secondary tables. If the individual's ID is used as the key variable, then each record in the secondary file has an ID, a start time, an end time, possibly a time origin, and a value for each of the TD covariates for the time interval specified by the times. An example of a secondary file is:

ID	StTime	EndTime	Weight	Cholesterol	# Name-record
1	2	2	2	2	# type-record
307	0	5	76.3	38.2	# 1st record for indiv 307
307	5	8	78.2	39.3	
307	8	12	81.7	39.7	
523	0	7	66.2	28.4	# 1st record for indiv 523
523	7	15	69.1	29.3	
- - -	- - - - -	etc. - - -	- - - -	- - - -	

Here there are two TD variables, Weight and Cholesterol, given on adjacent time-intervals for each individual. There can be as many records for an individual as necessary. In this example, the variable names and types are given in the data file.

In some cases, it may be more appropriate to use a different key variable. For example, a single air pollution monitor located in a city will be associated with every individual living in the city, and if the data give the readouts for monitors located in several cities, then these variables will be indexed by city rather than by individual. The ACS data set is structured like that. Table 1 is the beginning of the fictional secondary

file ACSlikeSec.dat, which is included in the test problems supplied with the program package.

The variable CITY corresponds to the same variable in the primary file, allowing the association to be made for each individual's city. The use of a secondary table can give a considerable saving in memory space, and sometimes a small saving in running time, compared with the same problem using only a primary file with time-dependence represented by “record-repeating”. In Table 1 below, the set of time-intervals for each city are the same; this is not necessary: the records for each city can divide up time in any way desired (although two intervals for the same city can not overlap). For saving both run-time and storage, it is advantageous to have as few time-intervals per individual as possible, consistent with a good approximation of the data.

The conceptual data matrix **M** is the same whether a secondary table is used or not: a secondary table is just a matter of how **M** is represented in the data files and in the computer. The algorithms for using a secondary table are different in detail from those used when no secondary table is supplied, but similar in essence.

Table 1: A Secondary Table

CITY	STTIME	ENDTIME	PM10	# Name-record
1	2	2	2	# type-record
1	0.0	7.5	0.4900	# start of data for city 1
1	7.5	15.0	0.5710	
1	15.0	22.5	0.3373	
1	22.5	30.0	0.8295	
1	30.0	37.5	0.7811	
1	37.5	45.0	0.8075	
2	0.0	7.5	0.5341	# start of data for city 2
2	7.5	15.0	0.8151	
2	15.0	22.5	0.7458	
2	22.5	30.0	0.5872	

2	30.0	37.5	0.4977	
2	37.5	45.0	0.7124	
3	0.0	7.5	0.5999	# start of data for city 3
3	7.5	15.0	0.7713	
3	15.0	22.5	0.4948	
3	22.5	30.0	0.7268	
- - - -	- - - -	etc. - - -	- - - -	

_Z_-Vectors

Given a pair $[e,\tau]$, with e an individual and τ an elapsed time: if e is at risk (and under observation) at τ, there is a unique *k* such that row *k* of **M** corresponds to $e = e^k$, and $\tau \in (\tau^k_{start}, \tau^k_{end}]$; denote this *k* value by $k(e,\tau)$. If e is not at risk at τ, define $k(e,\tau) = \infty$.

We define a conceptual index-set **Z**, first assuming there is no stratification: let **Z** denote the set of pairs $[e,\tau]$, where

1. e is at risk and under observation at τ.
2. $\tau = \tau_h$, for some *h*, $1 \le h \le q$. Here τ_h is one of the list *F* of event times defined earlier.

If there is stratification, then we define *Z* as the set of pairs $[e,\tau]$ such that

1. e is at risk and under observation at τ (i.e. $k[e,\tau] < \infty$)
2. $\tau = \tau_{sh}$, for some *h*, $1 \le h \le q_s$, where *s* is the stratum of $k[e,\tau]$.

The map $k[e,\tau]$ is many-to-one; the multiplicity $n(k)$ is defined as the number of pairs $[e,\tau]$ that map to the same value *k*:

$$n(k) = \#\{[e,\tau] \in Z \mid k(e,\tau) = k\} \tag{2}$$

This is the same as the number of τ_h that lie in $(\tau^k_{start}, \tau^k_{end}]$.

The set *Z* is ordered by τ within e within stratum. It is a conceptual index-set, which is never actually formed in the computation. Let $N_{\mathbf{z}}$ be the size of *Z*. Any vector which is indexed by Z is called a *Z*-vector, and normally considered a column vector. These *Z*-vectors are also never formed explicitly. We will write the entry of a *Z*-vector θ corresponding to $[e,\tau]$ as either $\theta[e,\tau]$ or $\theta^{[e,\tau]}$. Sometimes we denote pairs in *Z* by κ, and then we use the notation θ^{κ}. This convention also holds for matrices **A** whose columns are *Z*-vectors: $\mathbf{A}[e,\tau]$ or $\mathbf{A}^{[e,\tau]}$ or $\mathbf{A}^{\kappa}$ means the row of **A** corresponding to $\kappa=[e,\tau]$.

Given an extended vector $[\alpha^{\mathbf{T}}\ \beta^{\mathbf{T}}]^{\mathbf{T}}$ of regression coefficients (the α will be defined later, in section "Poisson models and Likelihood"), let the *Z*-vector μ be defined as

$$\mu[e,\tau_{sh}]=\mu^{[e,\tau_{sh}]}=\exp(\alpha^{sh}+\mathbf{R}^{k(e,\tau_{sh})}\beta)$$

for all $[e,\tau_{sh}]\in Z$. Another way of stating this is to define a conceptual matrix $\widetilde{\mathbf{R}}$ corresponding to $\mathbf{R}$: $\widetilde{\mathbf{R}}$ will have columns which are *Z*-vectors. Define the $[e,\tau]$ row of $\widetilde{\mathbf{R}}$ as

$$\widetilde{\mathbf{R}}^{[e,\tau]}=\mathbf{R}^{k(e,\tau)}$$

That is, the $[e,\tau]$ row of $\widetilde{\mathbf{R}}$ is the $k[e,\tau]$ row of $\mathbf{R}$. Then we can define μ simply as

$$\mu=\exp(\alpha)\circ\exp(\widetilde{\mathbf{R}}\beta)$$

where the exponential is applied entrywise, and "$\circ$" means the entrywise product, with the vector $\exp(\alpha)$ expanded to the size of a *Z*-vector, i.e

$$\alpha^{[e,\tau_{sh}]}=\alpha^{sh}$$

for all *s*, *h*, and e.

We now define the *Z*-vector Y as

$$Y[e,\tau] = Y^{[e,\tau]} = \begin{cases} 1 & \text{If } e \text{ has an event at time } \tau \\ 0 & \text{else} \end{cases}$$

for all $[e,\tau] \in Z$. Under the survivals models that are estimated, we have $E(Y) = \mu$. Notice that Y and χ contain essentially the same information, but they are vectors of (usually) different dimension. Another characterization of Y is

$$Y[e,\tau] = \begin{cases} 1 & \text{if } \chi^{k(e,\tau)} = 1 \;\&\; \tau = \tau_{end}^{k(e,\tau)} \\ 0 & \text{else} \end{cases}$$

For any cluster λ, and any *Z*-vector θ, define the *Z*-vector $\theta^{[\lambda]}$ as

$$\theta^{[\lambda]}[e,\tau] = \begin{cases} \theta[e,\tau] & \text{if } e \in i \\ 0 & \text{if } e \notin i \end{cases}$$

for all $[e,\tau] \in Z$. Then $\theta^{[\lambda]}$ is a *Z*-vector. It coincides with θ on the pairs corresponding to cluster λ, and is 0 in all other entries. We also define the sum of the entries of $\theta^{[\lambda]}$ as

$$\text{sum}(\theta^{[\lambda]}) = \sum \{\theta[e,\tau] |\; e \in \lambda\}$$

that is, the sum of $\theta[e,\tau]$ for all pairs $[e,\tau]$ such that $e \in \lambda$.

Summations of the form

$$\sum_{s=1}^{a} \sum_{h=1}^{q} \sum_{k \in R_{sh}} A(s,h,k)$$

over strata *s*, event-times τ_{sh} and members *k* of the risk set R_{sh}, occur frequently in the formulation of the problem. We note that such a sum is in fact a sum over *Z*, as follows:

$$\sum_{s=1}^{a} \sum_{h=1}^{q} \sum_{k \in R_{sh}} A(s,h,k) = \sum_{[e^k,\tau_{sh}] \in Z} A(s,h,k)$$

since $[e^k,\tau_{sh}] \in Z$ if and only if $k \in R_{sh}$.

As we will see in the next section, *Z*-vectors are fundamental in the formulation of the survival models; for example, the log-likelihood is a summation over a *Z*-vector. This leads to a difficulty: the dimension of any *Z*-vector is $N_{\mathbf{z}}$, which can be very large. In the ACS data, the number of individuals is about half a million, and the average number of distinct event times per stratum is about 90, so $N_{\mathbf{z}}$ is around 45 million. A naive algorithm for computing the log-likelihood and its derivatives requires stepping through *Z* on each iteration, a very time-consuming process. We describe below algorithms which avoid this in many important cases, and require stepping only through the rows of **M**. In many, if not most, problems, the number $N_{\mathbf{M}}$ of rows of **M** will be much smaller than $N_{\mathbf{z}}$. If there are no time-dependent covariates, for example, then $N_{\mathbf{M}}$ will be smaller than $N_{\mathbf{z}}$ by a factor equal to the average number of distinct event-times per stratum, about 90 for the ACS data.

Remark on Notation

As mentioned in the introduction, the algorithms given here depend on the statistical theory provided by Ma and colleagues (2000), here called "MKB"for brevity. We give here a brief explanation of how the notation of MKB differs from that used here. The main difference lies in the indexing of the set of individuals, and of the risk sets. For a 2-level random effects covariance model, MKB denotes by $x_{ijk}^{(s)}$, the vector of covariates associated with individual *k* in sub-cluster *j* of cluster *i*, in stratum *s*. The indexing of individuals is by cluster and subcluster, with the individual's ID *k* being assumed unique only within the subcluster. Here, in contrast, we denote the set of individuals by E, and usually index this set by e, which we think of as a globally unique identifier of the

individual (this does not imply that such an ID variable must be in the data: see below). We denote the full row vector of covariates (include the dummy covariates associated with the "alpha" coefficients) belonging to individual e, by $\mathbf{X}^e$, or $\mathbf{X}^e(\tau)$ in the case of time-dependent covariates, where τ denotes elapsed time. The original covariate vector, (a row vector) without the dummies, we denote by $\mathbf{R}^e$ or $\mathbf{R}^e(\tau)$. We denote the "leaf" cluster to which *e* belongs by *r(e)* and the random effect associated with a leaf cluster *r*, by U^r.

In a stratum *s*, we denote by τ_{sh} the distinct event times in that stratum, where *h* runs from 1 to q_s, the number of such times. This is the same notation as used by MKB. As in MKB, we denote the risk set at time τ by $R(\tau)$, and the risk set at time τ_{sh} by R_{sh}. The risk set is conceptually a set of individuals, but for practical purposes we take it as a set of records in the data file. In fact, all the computations are organized around data records, and individuals play no direct role; an ID variable for individuals is not needed, except possibly for grouping in a robust variance estimation. Each data record *k* is associated conceptually with an individual *e* and a time τ, and we write $k = k(e, \tau)$. Similarly we write *e(k)* to denote the individual associated with a data record *k*. Extending this scheme, we write $r^k = r(e(k))$, the leaf-cluster containing the individual belonging to record *k*; also $\mathbf{X}^k$ and $\mathbf{R}^k$, the extended and original covariate vectors associated with *e(k),* and U^{r^k}, the random effect associated with the leaf cluster containing the individual belonging to record *k*. The risk set R_{sh} is the set of data records belonging to the individuals at risk at time τ_{sh}. Using the notation scheme, we can write the conditional Poisson log-likelihood (see section Poisson Models and Likelihood) as:

$$\log(\ell(\alpha,\beta;Y\mid U))=\sum_{s=1}^{a}\sum_{h=1}^{q}\sum_{k\in R_{sh}}\left[\left(\log(U^{r^k})+\alpha^{sh}+\mathbf{R}^k\beta\right)\chi^k-U^{r^k}\exp(\alpha^{sh}+\mathbf{R}^k\beta)\right]$$

Here the summation is over strata *s*, event-times τ_{sh}, and risk-set members $k \in R_{sh}$, α^{sh} denotes the alpha-coefficient of stratum *s* and event time τ_{sh}, β is the coefficient vector corresponding to the covariates **R**, and χ^k is the event-indicator (0 or 1) of the data-record *k*. The same formula in the MKB notation is:

$$\log(\ell(\alpha,\beta;Y\mid \mathbf{U}))=\sum_{s=1}^{a}\sum_{h=1}^{q}\sum_{(i,j,k)\in R_{sh}}\left[\left(\log(u_{i,j})+\alpha_{sh}+x_{i,j,k}^{\mathrm{T}}\beta\right)Y_{i,j,k,h}^{(s)}-u_{i,j}\exp(\alpha_{sh}+x_{i,j,k}^{\mathrm{T}}\beta)\right]$$

Summation over all event-times and over their associated risk sets is equivalent to summation over the index-vector Z described above (section **`Z`** -Vectors). We use this equivalence frequently in what follows.

Log-Likelihood

Cox Model

We consider the Cox proportional hazards model

$$\lambda^e(\tau)=\lambda_0(\tau)\exp(R^e(\tau)\beta)$$

where $\lambda^e(\tau)$ is the hazard function for individual e, $\lambda_0(\tau)$ is the baseline hazard, common to all individuals in a stratum, β is a regression coefficient vector, and $R^e(\tau)$ is a row vector of covariate values for individual e and time τ.

No Ties, No Strata

Consider first the case of unstratified data, and no ties. We let $u^{r(e)}$ denote the random effect associated with the leaf-cluster $r(e)$ containing individual e. The log-likelihood

conditional on the random effects U is (Andersen and Gill 1982; Therneau and Grambsch 2000)

$$\log(p\ell(\beta; Y|\mathbf{U})) =$$
$$C + \sum_{e\in \mathrm{E}} \int_0^\infty \Big[\delta[e,\tau](\log(U^{r(e)}) + \widetilde{\mathbf{R}}^{[e,\tau]}\beta)$$
$$\log\left(\sum_{g\in \mathrm{E}} \delta[g,\tau]U^{r(g)} \exp(\widetilde{\mathbf{R}}^{[g,\tau]}\beta)\right)\Big] dN_e(\tau)$$

where $N_e(\tau)$ is the number of events suffered by individual e up to time τ. The factor of $\delta[e,\tau]$ on the first term inside the integral is redundant, since N_e cannot have a jump at τ unless $\delta[e,\tau]=1$. The constant C includes various terms, and is independent of the regression coefficients and the random effects. This expression is a summation over individuals and event-times, so in effect over Z. We can write it out explicitly as a sum over event times and the rows of **M**, interchanging the summations over individuals and times, giving

$$\log(p\ell(\beta; Y|\mathbf{U})) =$$
$$C + \sum_{h=1}^{q}\left[\sum_{k\in D_h}[\log(U^{r^k}) + \mathbf{R}^k\beta] - \log\left(\sum_{k\in R_h} U^{r^k}\exp(\mathbf{R}^k\beta)\right)\right] \quad (3)$$

With Ties

For tied data (still with no stratification), in which more than one event can occur at one time, we assume that the multiplicities are the result of grouping of continuous-time data. Then we have:

$$\log(p\ell(\beta;Y|\mathbf{U}))$$
$$= C + \sum_{e\in \mathrm{E}} \int_0^{\infty} \left[\delta[e,\tau](\log(U^{r(e)}) + \widetilde{\mathbf{R}}[e,\tau]\beta) - \sum_{j=1}^{m(\tau)} \log\left(\mathbb{P}^j(\tau,\beta,\mathbf{U})\right) \right] dN_e(\tau)$$

$$= C + \sum_{h=1}^{q} \left[\sum_{k\in D_h} [\log(U^{r^k}) + \mathbf{R}^k\beta] - \sum_{j=1}^{m_h} \log\left(\mathbb{P}^j(\tau_h,\beta,\mathbf{U})\right) \right]$$

where $m_h = m(\tau_h)$ is the multiplicity at time τ_h (the number of failures in the grouping interval around τ), and we have absorbed a term of log(m!) into the constant C; and for the Breslow-Peto approximation,

$$\mathbb{P}^j(\tau,\beta,\mathbf{U}) = \mathbb{P}(\tau,\beta,\mathbf{U}) = \sum_{g\in \mathrm{E}} \delta[g,\tau] U^{r(g)} \exp(\widetilde{\mathbf{R}}^{[g,\tau]}\beta)$$

$$= \sum_{k\in R(\tau)} U^{r^k} \exp(\mathbf{R}^k\beta)$$

so

$$\mathbb{P}_{sh}(\tau_{sh},\beta,\mathbf{U}) = \sum_{k\in R_{sh}} U^{r^k} \exp(\mathbf{R}^k\beta) \quad \text{(Breslow - Peto)}$$

Weighted: Let $\mathbf{w}$ be a case-weight vector. We then have

$$\log(p\ell(\beta;Y|\mathbf{U}))$$
$$= C + \sum_{h=1}^{q} \left[\sum_{k\in D_h} [\log(U^{r^k}) + \log(w^k) + \mathbf{R}^k\beta] - \sum_{j=1}^{m_h} \log\left(\mathbb{P}^j(\tau_h,\beta,\mathbf{U},\mathbf{w})\right) \right]$$

and

$$\mathbb{P}_{sh}(\tau_{sh},\beta,\mathbf{U},\mathbf{w}) = \sum_{k\in R_{sh}} U^{r^k} w^k \exp(\mathbf{R}^k\beta) \quad \text{(Breslow - Peto)}$$

So, for the Breslow-Peto approximation,

$$\log(p\ell(\beta;Y|\mathbf{U})) = C + \sum_{h=1}^{q} \left[\sum_{k\in D_h} [\log(U^{r^k}) + \log(w^k) + \mathbf{R}^k\beta] - m_h \log\left(\mathbb{P}(\tau_h,\beta,\mathbf{U},\mathbf{w})\right) \right]$$

For the Efron approximation,

$$\mathrm{P}^j(\tau,\beta,\mathbf{U}) = \sum_{g\ \in \mathbb{E}} \delta[g,\tau]\left[1+\left(\frac{j}{m(\tau)}-1\right)Y^{[g,\tau]}\right] U^{r(g)} \exp(\widetilde{\mathbf{R}}^{[g,\tau]}\beta)$$

$$= \sum_{k\ \in R(\tau)}\left[1+\left(\frac{j}{m(\tau)}-1\right)\chi^k\right] U^{r^k} \exp(\mathbf{R}^k\beta)$$

so

$$\mathrm{P}^j(\tau_h,\beta,\mathbf{U}) = \sum_{k\ \in R_h}\left[1+\left(\frac{j}{m_h}-1\right)\chi^k\right] U^{r^k} \exp(\mathbf{R}^k\beta) \quad \text{(Efron)}$$

where, as before, $Y[g,\tau]=1$ if g fails at time τ. In terms of the matrix **M**, this is equivalent to $\chi^k = 1$, where $k = k(g,\tau)$.

We can write the log-likelihood for the Breslow-Peto approximation as:

$$\begin{aligned}&\log(p\ell(\beta;Y|\mathbf{U}))\\ &= C + \sum_{h=1}^{q}\left[\sum_{k\ \in D_h}[\log(U^{r^k}) + \log(w^k) + \mathbf{R}^k\beta] - m_h \log\left(\mathrm{P}(\tau_h,\beta,\mathbf{U},\mathbf{w})\right)\right]\end{aligned} \tag{4}$$

As shown by Whitehead (1980) (see also Ma and colleagues (2000)), this is equivalent to a Poisson generalized linear model, which we describe in the next-but-one section. The Efron approximation does not seem to be equivalent to a Poisson model, or indeed to any GLM, so we will emphasize the Breslow-Peto approximation in what follows.

With Stratification

A stratum is a set of rows of **M**, and the log-likelihood is formed independently in each stratum; the separate stratum values are simply added. It follows that we can accommodate stratification simply by sorting the rows of **M** by τ_{start} within τ_{end} within stratum. Then each stratum *s* corresponds to a submatrix $\mathbf{M}^{[s]}$ of **M**, and we can simply sum up the log-likelihoods from the $\mathbf{M}^{[s]}$.

Poisson models and Likelihood

We introduce values $\{\alpha^{sh}\}$ for each stratum *s* and event-time τ_{sh}: we postulate that, given random effects $U = \mathbf{u}$, the values $Y[e, \tau_{sh}]$ are conditionally independent, and have conditional distribution

$$Y^{[e,\tau_{sh}]} | U \sim \text{Poisson}(U^{r(e)} \exp(\alpha^{sh} + \widetilde{\mathbf{R}}^{[e,\tau_{sh}]} \beta))$$

Weighted: if $\mathbf{w}$ is a weight vector, assumed to be a $\mathtt{z}$-vector,

$$Y^{[e,\tau_{sh}]} | U \sim \text{Poisson}(U^{r(e)} w^{[e,\tau_{sh}]} \exp(\alpha^{sh} + \widetilde{\mathbf{R}}^{[e,\tau_{sh}]} \beta))$$

So

$$\begin{aligned} \varepsilon(Y|(U) &= \mathbf{U} \circ \mathbf{w} \circ \mu \\ \varepsilon(Y) &= \varepsilon_{\mathbf{u}}(\varepsilon(Y|(U)) \\ &= \varepsilon_{\mathbf{u}}(\mathbf{U} \circ \mathbf{w} \circ \mu) = \mathbf{w} \circ \mu \end{aligned}$$

where the notation $\mathbf{w} \circ \mu$ means the entry-wise product of two vectors.

This is the same, letting $k = k(e, \tau_{sh})$, as

$$Y^{[e,\tau_{sh}]} | U \sim \text{Poisson}(U^{r^k} w^k \exp(\alpha^{sh} + \mathbf{R}^k \beta))$$

The conditional log-likelihood for this, given the random effects, (ignoring an additive constant) is

$$\begin{aligned} &\log(\ell(\alpha, \beta; Y | \mathbf{U})) = \\ &\sum_{s=1}^{a} \sum_{h=1}^{q} \sum_{k \in R_{sh}} \left[[\log(U^{r^k}) + \log(w^k) + \alpha^{sh} + \mathbf{R}^k \beta] \chi^k - U^{r^k} w^k \exp(\alpha^{sh} + \mathbf{R}^k \beta) \right] \\ &= \sum_{[e^k, \tau_{sh}] \in Z} \left[[\log(U^{r^k}) + \log(w^k) + \alpha^{sh} + \mathbf{R}^k \beta] Y^{[e^k, \tau_{sh}]} - U^{r^k} w^k \mu^{[e^k, \tau_{sh}]} \right] \end{aligned}$$

where Y and μ are *Z*-vectors defined above. With a little more notation, we can make this more compact. Recall the conceptual matrix $\widetilde{\mathbf{R}}$, whose columns are *Z*-vectors: we define another matrix $\widetilde{\mathbf{E}}$, with a column for each stratum-event-time pair sh, and whose columns are *Z*-vectors. The $[e,\tau_{sh}]$ row of $\widetilde{\mathbf{E}}$ has 1 in column sh, and 0 elsewhere; so $(\widetilde{\mathbf{E}}\boldsymbol{\alpha})^{[e,\tau_{sh}]} = \alpha^{sh}$. Now let

$$\widetilde{\mathbf{X}} = \begin{bmatrix} \widetilde{\mathbf{E}} & \widetilde{\mathbf{R}} \end{bmatrix}$$

and

$$\gamma = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Then we can write the log-likelihood as

$$\log(\ell(\alpha,\beta;Y\,|\,\mathbf{u})) = Y^{\mathbf{T}}[\log(U^*) + \log(\mathbf{w}) + \widetilde{\mathbf{X}}\gamma] - (\mathbf{U}^*)^{\mathbf{T}}(\mathbf{w}\circ\mu) \tag{5}$$

where $\mathbf{u}^*$ is the vector of leaf-level random effects, expanded into a Z-vector, i.e.

$$U^*[e^k,\tau_{sh}] = U^{r^k}$$

As mentioned above, this Poisson generalized linear model is equivalent to the Cox proportional hazards model using the Breslow-Peto approximation for ties. The proof is given in Whitehead (1980) and in Ma and colleagues (2000). Whitehead also gives (in formula 5.2) an interpretation of the α's: for each stratum *s*, the cumulative sum of $\exp(\alpha^{sj})$ up to $j = h$, is an estimate of the baseline cumulative hazard function for stratum *s*, at time τ_{sh}.

For a given random effects vector **U**, we estimate the regression coefficients α and β by maximizing the log-likelihood, which depends on α and β through the terms $Y^{\mathbf{T}}\widetilde{\mathbf{X}}\gamma$

and $(\mathbf{U}^*)^{\mathbf{T}}\mu$. This is easy for α: differentiating the log-likelihood with respect to α^{sh}, we find

$$\begin{aligned}\left(\partial/\partial\alpha^{sh}\right)\log(\ell(\alpha,\beta;Y|\,\mathbf{U})) &= \sum_{k\ \in R_{sh}}\left[\chi^k - u^{r^k}w^k\exp(\alpha^{sh}+\mathbf{R}^k\beta)\right]\\ &= m_{sh} - \exp(\alpha^{sh})\mathrm{P}_{sh}(\beta,\mathbf{U},\mathbf{w})\\ &\text{where}\\ \mathrm{P}_{sh}(\beta,\mathbf{U},\mathbf{w}) &= \sum_{k\ \in R_{sh}} U^{r^k}w^k\exp(\mathbf{R}^k\beta)\end{aligned}$$

and m_{sh} is defined above as the event count in stratum *s* at time τ_{sh}. It follows that

$$\exp(\alpha^{sh}) = m_{sh}\,.\left(\mathrm{P}_{sh}(\beta,\mathbf{U},\mathbf{w})\right.$$

so that α^{sh} can be determined as soon as β and the random effects are known.

Differentiating (5) with respect to γ gives

$$\partial\log(\ell(\alpha,\beta;Y|\,\mathbf{U}))/\partial\gamma = Y^{\mathbf{T}}\widetilde{\mathbf{X}} - (\mathbf{U}^*)^{\mathbf{T}}\,\mathrm{diag}(\mathbf{w}\circ\mu)\widetilde{\mathbf{X}}$$

whose transpose is

$$\Psi(\gamma|\mathbf{U}) = \nabla_\gamma\log(\ell(\alpha,\beta;Y|\,\mathbf{U})) = \widetilde{\mathbf{X}}^{\mathbf{T}}[Y - \mathrm{diag}(\mathbf{w}\circ\mu)(\mathbf{U}^*)] \tag{6}$$

where $\mathrm{diag}(\mathbf{w}\circ\mu)$ means the diagonal matrix whose diagonal is the vector $\mathbf{w}\circ\mu$. We now introduce more notation: let

$$A = \mathrm{diag}(\mathbf{w}\circ\mu)$$

and let B be a matrix whose columns are *Z*-vectors, with each column corresponding to a leaf cluster r. The column of cluster r is the *Z*-vector $(\mathbf{w}\circ\mu)^{[r]}$, defined above as equal to $\mathbf{w}\circ\mu$ at all positions $[e^k,\tau_{sh}]$ for which $e^k\in r,$ and 0 elsewhere; that is,

$$B_r^{[e^k,\tau_{sh}]} = \begin{cases} w^k\mu^{[e^k,\tau_{sh}]}, & e^k\in r\\ 0, & e^k\notin r\end{cases}$$

In this notation, we have

$$\text{diag}(\mathbf{w} \circ \mu)\mathbf{U}^* = B\,\mathbf{U}$$

and

$$\Psi(\gamma \,|\, \mathbf{U}) = \nabla_\gamma \log(\ell(\alpha, \beta; Y \,|\, \mathbf{U})) = \widetilde{\mathbf{X}}^{\mathbf{T}}[Y - B\,\mathbf{U}]$$

Equivalence with Cox Model

Now as before, letting

$$\mathbb{P}(\tau_{sh}, \beta, \mathbf{U}, \mathbf{w}) = \sum_{k \in R_{sh}} U^{r^k} w^k \exp(\mathbf{R}^k \beta)$$

we have

$$\log(\ell(\alpha, \beta; Y \,|\, \mathbf{u})) = \sum_{s=1}^{a}\sum_{h=1}^{q}\left[\sum_{k \in R_{sh}} [\log(u^{r^k}) + \log(w^k) + \alpha^{sh} + \mathbf{R}^k \beta]\,\chi^k - \exp(\alpha^{sh})\mathbb{P}(\tau_{sh}, \beta, \mathbf{u}, \mathbf{w})\right]$$

The value of α at the maximum is determined by

$$\begin{aligned} 0 = \partial \log(\ell(\alpha, \beta; Y \,|\, \mathbf{U})) / \partial \alpha^{sh} &= \sum_{k \in R_{sh}} \chi^k - \exp(\alpha^{sh})\mathbb{P}(\tau_{sh}, \beta, \mathbf{U}, \mathbf{w}) \\ &= m_{sh} - \exp(\alpha^{sh})\mathbb{P}(\tau_{sh}, \beta, \mathbf{U}, \mathbf{w}) \end{aligned}$$

Setting this to zero gives

$$\begin{aligned} \exp(\alpha^{sh}) &= m_{sh} \,/\, \mathbb{P}(\tau_{sh}, \beta, \mathbf{U}, \mathbf{w}) \\ \alpha^{sh} &= \log(m_{sh} \,/\, \mathbb{P}(\tau_{sh}, \beta, \mathbf{U}, \mathbf{w})) = g^{sh}(\beta, \mathbf{U}, \mathbf{w}) \\ &\text{or} \\ \alpha &= g(\beta, \mathbf{U}, \mathbf{w}) \end{aligned} \tag{7}$$

say. This is the value of α^{sh} at the maximum of $\log(\ell(\alpha, \beta; Y \,|\, \mathbf{U}))$.

Substituting this for α^{sh} in $\log(\ell(\alpha, \beta; | \,\mathbf{U}))$ gives

$$\begin{aligned} \log(\ell(\alpha, \beta; Y \,|\, \mathbf{U})) = \sum_{s=1}^{a}\sum_{h=1}^{q}\Bigg[\sum_{k \in \mathbf{R}_{sh}} [\log(U^{r^k}) + \log(w^k) + \alpha^{sh} + \mathbf{R}^k \beta]\,\chi^k - \\ \exp(\alpha^{sh})\mathbb{P}(\tau_{sh}, \beta, \mathbf{U}, \mathbf{w})\Bigg] \end{aligned}$$

$$= \sum_{s=1}^{a}\sum_{h=1}^{q}\left[\sum_{k\ \in R_{sh}} [\log(U^{r^k}) + \log(w^k) + \log(m_{sh}) - \log(\mathrm{P}(\tau_{sh}, \beta, \mathbf{U}, w)) + \mathbf{R}^k\beta]\chi^k - m_{sh}\right]$$

$$= C_1(Y) + \sum_{s=1}^{a}\sum_{h=1}^{q}\left[\sum_{k\ \in R_{sh}} [\log(U^{r^k}) + \log(w^k) - \log(\mathrm{P}(\tau_{sh}, \beta, \mathbf{U}, w)) + \mathbf{R}^k\beta]\chi^k\right]$$

$$= C_1(Y) + \sum_{s=1}^{a}\sum_{h=1}^{q}\left[\sum_{k\ \in R_{sh}} [\log(U^{r^k}) + \log(w^k) + \mathbf{R}^k\beta]\chi^k - \sum_{k\ \in R_{sh}} \chi^k \log(\mathbb{P}(\tau_{sh}, \beta, \mathbf{U}, \mathbf{w}))\right]$$

$$= C_1(Y) + \sum_{s=1}^{a}\sum_{h=1}^{q}\left[\sum_{k\ \in \mathbf{D}_{sh}} [\log(U^{r^k}) + \log(w^k) + \mathbf{R}^k\beta] - m_{sh}\log(\mathrm{P}(\tau_{sh}, \beta, \mathbf{U}, w))\right]$$

where

$$C_1(Y) = \sum_{s=1}^{a}\sum_{h=1}^{q}\left[\sum_{k\ \in R_{sh}} \log(m_{sh})\chi^k - m_{sh}\right]$$

Comparing the expression for $\log(\ell(\alpha, \beta; Y \mid \mathbf{u}))$ given here, with Therneau and Li (1998), we see they differ by an expression independent of β and $\mathbf{u}$, when strata are allowed for. It follows that the Poisson model gives identical estimates of β, to the Cox model of Therneau and Li (1998). Unfortunately, it appears no such argument is possible for the Efron approximation.

Derivatives of $\log(\ell)$

We have, letting $\mathbf{z}$ be an offset variable,

$$L = \log(\ell(\alpha, \beta; Y \mid \mathbf{U})) = \sum_{s=1}^{a} \sum_{h=1}^{q} \left[\sum_{k \in R_{sh}} [\log(u^{r^k}) + \log(w^k) + \alpha^{sh} + \mathbf{R}^k \beta + \mathbf{z}^k] \chi^k - \exp(\alpha^{sh}) \mathrm{P}_{sh}(\beta, \mathbf{U}, \mathbf{w}, \mathbf{z}) \right]$$

where

$$\mathrm{P}_{sh}(\beta, \mathbf{U}, \mathbf{w}, \mathbf{z}) = \sum_{k \in R_{sh}} u^{r^k} w^k \exp(\mathbf{R}^k \beta + \mathbf{z}^k)$$

The various derivatives of L are given in the table below, where a subscript α, β, or U means a partial derivative with respect to the variable in the subscript. Recall that $\mathbf{R}^k$ is a row vector, row k of $\mathbf{R}$, and α, β, and U are column vectors. The notation $\{L_\alpha\}^{sh}$, for example, means the *sh*-component of the gradient vector $L_\alpha = \nabla_\alpha L$, or in other words $\{L_\alpha\}^{sh} = \partial L / \partial \alpha^{sh}$. Here *sh* means stratum *s* and stratum *s*'s event-time *h*.

$$\{L_\alpha\}^{sh} = \{\partial \log(\ell(\alpha, \beta; Y \mid \mathbf{U})) / \partial \alpha\}_{sh} = m_{sh} - e^{\alpha^{sh}} \mathrm{P}_{sh}(\beta, \mathbf{U}, \mathbf{w}, \mathbf{z})$$

$$L_\beta = \partial \log(\ell(\alpha, \beta; Y \mid \mathbf{U})) / \partial \beta = \sum_{s=1}^{a} \sum_{h=1}^{q} \left[\sum_{k \in R_{sh}} \chi^k \mathbf{R}^k - \exp(\alpha^{sh}) \sum_{k \in R_{sh}} U^{r^k} w^k \exp(\mathbf{R}^k \beta + \mathbf{z}^k) \mathbf{R}^k \right]$$

$$= \sum_{s=1}^{a} \sum_{h=1}^{q} \left[\sum_{k \in R_{sh}} \chi^k \mathbf{R}^k - m_{sh} \frac{\sum_{\ell \in \mathbf{R}_{sh}} u^{r^\ell} w^\ell \exp(\mathbf{R}^\ell \beta + \mathbf{z}^\ell) \mathbf{R}^\ell}{\sum_{i \in R_{sh}} U^{r^i} w^i \exp(\mathbf{R}^i \beta + \mathbf{z}^i)} \right] \text{ at max}$$

$$= \sum_{s=1}^{a} \sum_{h=1}^{q} \left[\sum_{k \in \mathbf{R}_{sh}} \chi^k \mathbf{R}^k - m_{sh} \overline{\mathbf{R}}_{sh}(\beta) \right]$$

where

$$\overline{\mathbf{R}}_{sh}(\beta) = \frac{\sum_{\ell \in R_{sh}} U^{r^\ell} w^\ell \exp(\mathbf{R}^\ell \beta + \mathbf{z}^\ell) \mathbf{R}^\ell}{\sum_{i \in R_{sh}} U^{r^i} w^i \exp(\mathbf{R}^i \beta + \mathbf{z}^i)} = \frac{\sum_{\ell \in \mathbf{R}_{sh}} u^{r^\ell} w^\ell \exp(\mathbf{R}^\ell \beta + \mathbf{z}^\ell) \mathbf{R}^\ell}{\mathrm{P}_{sh}(\beta, \mathbf{U}, \mathbf{w}, \mathbf{z})}$$

$$\{L_{\alpha\alpha}\}^{sh}_{sh} = \{\partial^2 \log(\ell(\alpha, \beta; Y \mid \mathbf{U})) / \partial \alpha^2\}_{sh,sh} = -e^{\alpha^{sh}} \mathrm{P}_{sh}(\beta, \mathbf{U}, \mathbf{w}, \mathbf{z}) = -m_{sh} \quad \text{at max (matrix is diagonal)}$$

$$\{L_{\alpha\beta}\}^{sh} = \partial^2 \log(\ell(\alpha, \beta; Y \mid \mathbf{U})) / \partial \alpha \partial \beta = -e^{\alpha^{sh}} \partial \mathrm{P}_{sh}(\beta, \mathbf{U}, \mathbf{w}, \mathbf{z}) / \partial \beta = -e^{\alpha^{sh}} \sum_{k \in R_{sh}} U^{r^k} w^k \exp(\mathbf{R}^k \beta + \mathbf{z}^k) \mathbf{R}^k \quad (sh\text{-row})$$

$$L_{\beta\beta} = \partial^2 \log(\ell(\alpha, \beta; Y \mid \mathbf{U})) / \partial\beta^2 = -\sum_{s=1}^{a}\sum_{h=1}^{q} e^{\alpha^{sh}} \sum_{k \in R_{sh}} U^{r^k} w^k \exp(\mathbf{R}^k \beta + \mathbf{z}^k)(\mathbf{R}^k)^{\mathrm{T}} \mathbf{R}^k$$

$$\{\partial \mathbb{P}_{sh}(\beta, \mathbf{U}, \mathbf{w}, \mathbf{z}) / \partial \mathbf{U}\}^r = \sum_{k \in R_{sh} \,\&\, r^k = r} w^k \exp(\mathbf{R}^k \beta + \mathbf{z}^k)$$

$$\{L_{\mathbf{u}}\}^r = m^r / U^r - \mathrm{sum}((\mathbf{w} \circ \mu)^{[r]})$$

where $m^r = \mathrm{sum}(\chi^{[r]})$, the event count for leaf - cluster r

$$\{L_{\alpha\mathbf{u}}\}^r_{sh} = \partial^2 \log(\ell(\alpha, \beta; Y \mid \mathbf{U})) / \partial\alpha\partial\mathbf{U} = -e^{\alpha^{sh}} \{\partial \mathbb{P}_{sh}(\beta, \mathbf{U}, \mathbf{w}, \mathbf{z}) / \partial \mathbf{U}\}^r = -e^{\alpha^{sh}} \sum_{k \in R_{sh} \,\&\, r^k = r} w^k \exp(\mathbf{R}^k \beta + \mathbf{z}^k)$$

$$\{L_{\beta\mathbf{u}}\}^r = \{\partial^2 \log(\ell(\alpha, \beta; Y \mid \mathbf{U})) / \partial\beta\partial\mathbf{U}\}^r = \sum_{s=1}^{a}\sum_{h=1}^{q}\left[-e^{\alpha^{sh}} \sum_{k \in \mathbf{R}_{sh} \,\&\, r^k = r} w^k \exp(\mathbf{R}^k \beta + \mathbf{z}^k)\mathbf{R}^k \right] \quad (r^{th} \text{ column})$$

For future reference, we note the following: let the *sh*-row of $L_{\alpha\beta}$ be denoted by G_{sh}:

$$G_{sh} = -e^{\alpha^{sh}} \sum_{k \in R_{sh}} U^{r^k} w^k \exp(\mathbf{R}^k \beta + \mathbf{z}^k)\mathbf{R}^k$$

then, if (α, β) maximize L for given U,

$$L_{\beta\alpha}(L_{\alpha\alpha})^{-1} L_{\alpha\beta} = -\sum_{s=1}^{a}\sum_{h=1}^{q} \frac{1}{m_{sh}} (G_{sh})^{\mathrm{T}} G_{sh} \tag{8}$$

BLUP Predictors

Definitions and Properties

Now we consider the method of predicting the components of the random effect vector *U*. We recall that N_{leaf} is the total number of leaves, which are ordered lexically. For this section we will assume that values (α, β) are given, which in turn determine the *Z*-vector μ. The matrices *A* and *B* were defined above as $A = \mathrm{diag}(\mathbf{w} \circ \mu)$, and *B* is $N_{\mathbf{z}} \times N_{leaf}$, with column r of *B* being the *Z*-vector $(\mathbf{w} \circ \mu)^{[r]}$ (recall that $v^{[r]}$ is equal to v on pairs $[e, \tau]$ such that $e \in r$, and 0 elsewhere, for any *Z*-vector v). We note that the matrix

$$Q = B^{\mathrm{T}} A^{-1} B$$

is diagonal; it plays an important role in predicting U.

Expressions for $\mathrm{var}(Y)$ **and** $\mathrm{cov}(U, Y)$

As mentioned before, the clusters of any level k are disjoint. By a leaf-vector we mean a vector of dimension N_{leaf} whose components are associated with the corresponding leaves. The random-effects vector U is a leaf-vector. We start from the formulas

$$\begin{aligned} \mathrm{var}(Y) &= A + B\,\mathrm{var}(U)\,B^{\mathbf{T}} \\ &= A + B\,D\,B^{\mathbf{T}} \\ &\text{and} \\ \mathrm{cov}(U,Y) &= \mathrm{var}(U)B^{\mathbf{T}} = DB^{\mathbf{T}} \\ \varepsilon(Y) &= \mathbf{w} \circ \mu \end{aligned} \tag{9}$$

where $\mathrm{var}(Y) = \mathrm{cov}(Y, Y)$, $D = \mathrm{var}(U)$, $\mathbf{w}$ is a weight vector, and "$\circ$" means the entry-wise product of vectors. We note here that $\mathrm{var}(Y)$ is $N_{\mathbf{z}} \times N_{\mathbf{z}}$, and $\mathrm{var}(U)$ is $N_{leaf} \times N_{leaf}$. It is easy to see that if the weight vector $\mathbf{w}$ has positive entries, then B has full rank, which implies that $D = \mathrm{var}(U)$ is uniquely determined by $\mathrm{var}(Y)$.

We will assume here that $D = \mathrm{var}(U)$ is simply given; in fact, we need to have a model for $\mathrm{var}(U)$, and the available models normally contain parameters, called "dispersion parameters", which must be estimated as part of the overall estimation process. We will denote a particular covariance model by $D(\eta)$, where η is the vector of dispersion parameters, whose dimension and nature are specific to the covariance model considered. It is assumed that $D()$ is a known function, and that any admissible value of η determines the covariance matrix $D(\eta)$.

BLUP Formula

The BLUP predictor of U has the form $\hat{U} = m + HY$, where the vector m and matrix H are to be determined. The defining requirements are that $\varepsilon(\hat{U}) = \varepsilon(U) = 1$, and that $\hat{U} - U$ be orthogonal to any linear transformation GY of Y, for any matrix G. This implies

$$\varepsilon(\hat{U}) = m + H\varepsilon(Y) = m + H(\mathbf{w} \circ \mu) = 1$$

so

$$m = 1 - H(\mathbf{w} \circ \mu)$$

(note that 1 here is a leaf-vector whose entries are all 1), and

$$\operatorname{cov}(\hat{U} - U, GY) = 0$$

or

$$\operatorname{cov}(HY, GY) = \operatorname{cov}(U, GY)$$

so

$$H \operatorname{var}(Y) G^{\mathrm{T}} = \operatorname{cov}(U, Y) G^{\mathrm{T}}, \text{all } G,$$

$$\text{so} \quad H = \operatorname{cov}(U, Y) \operatorname{var}(Y)^{-1}$$

It follows that

$$\hat{U} = 1 + \operatorname{cov}(U, Y) \operatorname{var}(Y)^{-1} (Y - \mathbf{w} \circ \mu)$$

This is the BLUP formula. Using (9) and the Sherman-Morrison-Woodbury formula,

$$(A + WW^{\mathrm{T}})^{-1} = A^{-1} - A^{-1} W (I + W^{\mathrm{T}} A^{-1} W)^{-1} W^{\mathrm{T}} A^{-1}$$

we can determine the matrix H by

$$\begin{aligned}
\mathrm{var}(Y)^{-1} &= (A + B\,D\,B^{\mathbf{T}})^{-1} = (A + WW^{\mathbf{T}})^{-1} \\
&\text{where} \\
W &= BD^{1/2} \\
&\text{so} \\
\mathrm{var}(Y)^{-1} &= A^{-1} - A^{-1}W(I + W^{\mathbf{T}}A^{-1}W)^{-1}W^{\mathbf{T}}A^{-1} \\
&= A^{-1} - A^{-1}B(D^{-1} + Q)^{-1}B^{\mathbf{T}}A^{-1} \\
&= A^{-1} - A^{-1}BQ^{-1}(Q^{-1} + D)^{-1}DB^{\mathbf{T}}A^{-1} \\
&= A^{-1} - A^{-1}B(I + DQ)^{-1}DB^{\mathbf{T}}A^{-1} \\
&\text{and} \\
H = \mathrm{cov}(U,Y)\,\mathrm{var}(Y)^{-1} &= DB^{\mathbf{T}}\left[A^{-1} - A^{-1}BQ^{-1}(Q^{-1} + D)^{-1}DB^{\mathbf{T}}A^{-1}\right] \\
&= DB^{\mathbf{T}}A^{-1} - D(Q^{-1} + D)^{-1}DB^{\mathbf{T}}A^{-1} \\
&= [I - D(Q^{-1} + D)^{-1}]DB^{\mathbf{T}}A^{-1} \\
&= Q^{-1}(Q^{-1} + D)^{-1}DB^{\mathbf{T}}A^{-1}
\end{aligned}$$

The predicted random effect is

$$\begin{aligned}
\hat{U} &= 1 + Q^{-1}(Q^{-1} + D)^{-1}DB^{\mathbf{T}}A^{-1}(Y - \mu) \\
&= 1 + Q^{-1}(Q^{-1} + D)^{-1}D\,w \\
&= 1 + (I + DQ)^{-1}D\,w \\
&= 1 + \mathrm{BLUP}\,w
\end{aligned}$$

where

$$\begin{aligned}
Q &= B^{\mathbf{T}}A^{-1}B \\
\mathrm{BLUP} &= (I + DQ)^{-1}D \\
&= D(Q^{-1} + D)^{-1}Q^{-1} \\
&= Q^{-1}(Q^{-1} + D)^{-1}D \\
&\text{and} \\
w &= B^{\mathbf{T}}A^{-1}(Y - \mathbf{w} \circ \mu)
\end{aligned}$$

We note that Q is a diagonal matrix, whose diagonal entry corresponding to the leaf-cluster r is

$$Q_r^r = (B^{\mathbf{T}}A^{-1}B)_r^r = \mathrm{sum}((\mathbf{w} \circ \mu)^{[r]})$$

and

$$
\begin{aligned}
w &= B^{\mathbf{T}} A^{-1}(Y - \mathbf{w} \circ \mu) \\
w^r &= \text{sum}((Y - \mathbf{w} \circ \mu)^{[r]}) \\
\text{var}(w) &= Q + Q\,D\,Q
\end{aligned}
$$

Clearly,

$$
\begin{aligned}
\text{var}(\hat{U}) &= H\,\text{var}(Y)H^{\mathbf{T}} = DB^{\mathbf{T}}\,\text{var}(Y)^{-1}BD \\
&= (I + DQ)^{-1} DB^{\mathbf{T}} A^{-1} BD \\
&= (I + DQ)^{-1} DQD \\
&= [I - (I + DQ)^{-1}]D \\
&= D - (I + DQ)^{-1} D \\
\text{var}(\hat{U}) &= DQ(I + DQ)^{-1} D \\
&= DQ(Q + QDQ)^{-1} QD
\end{aligned}
$$

and

$$
\text{cov}(\hat{U}, U) = \text{cov}(\hat{U}, \hat{U}) = \text{var}(\hat{U})
$$

We can also write this relation in the form

$$
\begin{aligned}
D = \text{var}(U) &= \text{var}(\hat{U}) + (I + DQ)^{-1} D \\
&= \text{var}(\hat{U}) + \text{BLUP}
\end{aligned}
\tag{10}
$$

which can be considered a bias-correction formula.

Estimation of β

Basic Formulas

It is shown above that the Poisson model gives the same estimates for the regression coefficients β as does the Cox model with the Breslow-Peto approximation for ties. We develop here a method for estimating β from the Poisson model, which produces simultaneously the BLUP-predicted random effects $\hat{U}$. Recall that the conditional log-likelihood and its gradient are given by

$$\log(\ell(\alpha,\beta;Y\,|\,U)) = Y^{\mathbf{T}}[\log(U^*) + \log(\mathbf{w}) + \widetilde{\mathbf{X}}\gamma] - (U^*)^{\mathbf{T}}(\mathbf{w}\circ\mu)$$

$$\Psi(\gamma|U) = \nabla_\gamma \log(\ell(\alpha,\beta;Y\,|\,U)) = \widetilde{\mathbf{X}}^{\mathbf{T}}[Y - \mathrm{diag}(\mathbf{w}\circ\mu)U^*]$$

$$= \widetilde{\mathbf{X}}^{\mathbf{T}}[Y - AU^*] = \widetilde{\mathbf{X}}^{\mathbf{T}}[Y - BU]$$

where we note in passing that $AU^* = BU$.

The conditional expectation of $\Psi(\gamma|U)$ over U, given Y, is

$$\varepsilon_U(\Psi(\gamma|U)|Y) = \widetilde{\mathbf{X}}^{\mathbf{T}}[Y - B\,\varepsilon(U\,|\,Y)]$$

We approximate $\varepsilon(U\,|\,Y)$ by the BLUP predictor, defining a function $\psi(\gamma)$:

$$\psi(\gamma) = \psi(\gamma, Y) = \widetilde{\mathbf{X}}^{\mathbf{T}}[Y - B\hat{U}]$$

This is also equal to

$$\psi(\gamma) = \widetilde{\mathbf{X}}^{\mathbf{T}} A\,\mathrm{var}(Y)^{-1}(Y - \mu)$$

as can be proved by a simple argument. We will estimate γ by the equation $\psi(\gamma) = 0$.

Derivatives

Now we need some differentiation formulas:

$$\partial(\mathbf{w}\circ\mu)/\partial\gamma = \mathrm{diag}(\mu)\widetilde{\mathbf{X}} = A\widetilde{\mathbf{X}}$$

$$(\partial/\partial\gamma)(\mathbf{w}\circ\mu)^{[e,\tau]} = (\mathbf{w}\circ\mu)^{[e,\tau]}\,\widetilde{\mathbf{X}}^{[e,\tau]}$$

$$(\partial/\partial\gamma)(\mathbf{w}\circ\mu)^{[e,\tau]}(\widetilde{\mathbf{X}}^{[e,\tau]})^{\mathbf{T}} = (\widetilde{\mathbf{X}}^{[e,\tau]})^{\mathbf{T}}(\mathbf{w}\circ\mu)^{[e,\tau]}\,\widetilde{\mathbf{X}}^{[e,\tau]}$$

$$(\partial/\partial\gamma)\psi[d\gamma] = -\widetilde{\mathbf{X}}^{\mathbf{T}} A\,\mathrm{var}(Y)^{-1} A\,\widetilde{\mathbf{X}}[d\gamma] + \widetilde{\mathbf{X}}^{\mathbf{T}}\{\partial A/\partial\gamma\}[d\gamma]\,\mathrm{var}(Y)^{-1}(Y - \mu)$$

$$\varepsilon(\partial\psi/\partial\gamma) = -\widetilde{\mathbf{X}}^{\mathbf{T}} A\,\mathrm{var}(Y)^{-1} A\,\widetilde{\mathbf{X}} = -\mathbf{S}(\gamma)$$

That is, we let $\mathbf{S}(\gamma)$ denote the expectation (over Y) of the derivative of $\psi(\gamma)$ with respect to γ. If there is stratification,

$$\mathbf{S}(\gamma) = \varepsilon(\partial\psi / \partial\gamma) = \sum_{s=1}^{a} \mathbf{S}_s(\gamma_s)$$

Least Squares Equivalence

Define

$$\begin{aligned} q(\mu) &= (1/2)(Y - (\mathbf{w} \circ \mu))^{\mathbf{T}} \operatorname{var}(Y)^{-1} (Y - (\mathbf{w} \circ \mu)) \\ &= (1/2) Z^{\mathbf{T}} Z \end{aligned}$$

where Z is defined as

$$\operatorname{var}(Y)^{-1/2} (Y - (\mathbf{w} \circ \mu))$$

Then

$$\partial q / \partial \mu = (Y - (\mathbf{w} \circ \mu))^{\mathbf{T}} \operatorname{diag}(\mathbf{w}) \operatorname{var}(Y)^{-1}$$

It follows that

$$\begin{aligned} \psi(\gamma) &= [(Y - \mu)^{\mathbf{T}} \operatorname{var}(Y)^{-1} A \mathbf{X}]^{\mathbf{T}} \\ &= ((\partial q / \partial \mu)(\partial \mu / \partial \gamma))^{\mathbf{T}} = (\partial / \partial \gamma) q(\mu(\gamma)) \end{aligned}$$

so that the condition $\psi(\gamma) = 0$ is equivalent to minimizing $q(\mu(\gamma))$ with respect to γ, and amounts to a nonlinear least-squares criterion.

Estimating Equations

The basic estimating equations are then $\psi(\gamma) = 0$ and the BLUP formula for U:

$$\begin{aligned} \psi(\gamma) &= \tilde{\mathbf{X}}^{\mathbf{T}} [Y - B\hat{U}] = 0 \\ \hat{U} &= \mathbf{1} + (I + DQ)^{-1} DB^{\mathbf{T}} A^{-1} (Y - \mu) \end{aligned} \tag{11}$$

Note that these are stated in terms of the conceptual vectors and matrices Y, $\widetilde{\mathbf{X}}$, μ, A, B; we still must develop computational algorithms in terms of the data matrix $\mathbf{M}$.

We note that the BLUP formula for $\hat{U}$ depends on $D = \text{var}(U)$, which in turn depends on the dispersion parameters of the particular model for $\text{var}(U)$ that is used. To the estimating equations must be added one more, for determining the dispersion parameter vector η of the covariance model $D(\eta)$. Let us suppose that the dispersion parameter vector η is estimated with the help of an estimating equation

$$G(\gamma, \eta, U) = 0$$

or, in an equivalent fixed-point form,

$$\eta = H(\gamma, \eta, U)$$

and the full system of estimating equations becomes

$$\begin{aligned}
&\psi(\gamma) = \widetilde{\mathbf{X}}^{\mathbf{T}}[Y - B\hat{U}] = 0 && \text{(Score equation for } \gamma) \\
&\hat{U} = \mathbf{1} + (I + DQ)^{-1} DB^{\mathbf{T}} A^{-1}(Y - \mu) && \text{(BLUP predictor of } \hat{U}) \\
&D = D(\eta) && \text{(Model for var}(U)) \\
&\eta = H(\gamma, \eta, \hat{U}) && \text{(Estimation of dispersion parameters)}
\end{aligned}$$

Solving for α and β

Newton-Picard Iteration

Given the values of $\hat{U}$, the Fisher scoring algorithm for solving the first of the estimating equations above is given by

$$\mathbf{S}(\gamma_{old})(\gamma_{new} - \gamma_{old}) = -\psi(\gamma_{old}) \tag{12}$$

Now recall that the matrix $\widetilde{\mathbf{X}}$ is partitioned into "alpha" and "beta" columns by

$$\widetilde{\mathbf{X}} = \begin{bmatrix} \widetilde{\mathbf{E}} & \widetilde{\mathbf{R}} \end{bmatrix}$$

With this partitioning, we can write the Fisher scoring equation as

$$\begin{bmatrix} -M & -N \\ -N^{\mathbf{T}} & -P \end{bmatrix} \begin{bmatrix} \alpha_{new} - \alpha_{old} \\ \beta_{new} - \beta_{old} \end{bmatrix} = -\begin{bmatrix} p \\ q \end{bmatrix}$$

or

$$\begin{bmatrix} M & N \\ N^{\mathbf{T}} & P \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta\beta \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix}$$

where

$$\begin{aligned} M &= \widetilde{\mathbf{E}}^{\mathbf{T}} A \operatorname{var}(Y)^{-1} A \widetilde{\mathbf{E}} \\ N &= \widetilde{\mathbf{E}}^{\mathbf{T}} A \operatorname{var}(Y)^{-1} A \widetilde{\mathbf{R}} \\ P &= \widetilde{\mathbf{R}}^{\mathbf{T}} A \operatorname{var}(Y)^{-1} A \widetilde{\mathbf{R}} \end{aligned}$$

and

$$\begin{aligned} p &= \widetilde{\mathbf{E}}^{\mathbf{T}} [Y - B\hat{U}] \\ q &= \widetilde{\mathbf{R}}^{\mathbf{T}} [Y - B\hat{U}] \end{aligned}$$

Practicalities

The partitioned form of the score equation indicates a practical difficulty: the vector α can be large. For the ACS data, there can be around 200 strata, averaging nearly 90 distinct event-times each (because of ties, the average number of event-times per stratum is only weakly dependent on the fineness of the stratification). This means that the matrix M above is about $18,000 \times 18,000$, and would require over 2.5 gigabytes to store. We need a way to avoid forming this matrix or any of similar size. Since the α-vector is in some sense a nuisance parameter, and the real focus is on the regression coefficients β, we try eliminating $\Delta\alpha$ from the partitioned system. The result is

$$[P - N^{\mathbf{T}} M^{-1} N] \Delta\beta = q - N^{\mathbf{T}} M^{-1} p$$

Now we have already seen that the α-component of $\psi(\gamma)$ is the vector $p = \widetilde{\mathbf{E}}^{\mathbf{T}}[Y - B\hat{U}]$, the gradient of the log-likelihood L with respect to α, and from section "Derivatives of log(ℓ)" this is given by

$$p^{sh} = \{\widetilde{\mathbf{E}}^{\mathbf{T}}[Y - B\hat{U}]\}^{sh} = m_{sh} - e^{\alpha^{sh}} \mathtt{P}_{sh}(\beta, \hat{U}) \tag{13}$$

If we then choose each α^{sh} to satisfy

$$e^{\alpha^{sh}} = m_{sh} \,/\, \mathtt{P}_{sh}(\beta, \hat{U})$$

then $p = 0$ for this choice, and the equation (13) simplifies to

$$[P - N^{\mathbf{T}} M^{-1} N]\Delta\beta = q$$

The matrix $K = P - N^{\mathbf{T}} M^{-1} N$ is called the Schur complement of the original matrix $\mathbf{S}$, and it is the matrix needed to determine the new regression coefficients β on each iteration. As we shall see later, it is possible and practical to compute the Schur complement exactly, but the computation is time-consuming, and computing it on every iteration is too slow. It can be computed once, at the end, to determine standard errors, but for the iterations, we must find an approximation. For approximating, we have two choices: either compute the exact Schur complement and re-use it for several iterations to amortize the cost, or use an approximate Schur complement $\hat{K}$ for the iterations, then the exact K for standard errors. At present, the program uses the second alternative. The approximation to K used is the matrix

$$\hat{K} = L_{\beta\beta} - L_{\beta\alpha}(L_{\alpha\alpha})^{-1} L_{\alpha\beta}$$

Formulas for the terms are given in section "Derivatives of log(ℓ)", and $\hat{K}$ is reasonably fast to compute. The matrix $\hat{K}$ differs from K in that $\hat{K}$ is formed treating

the random effects vector $\hat{U}$ as a constant in the derivatives; that is, the derivatives with respect to α and β in $\hat{K}$ do not take account of the variation of $\hat{U}$ with α and β. It works adequately for the iterations, but is not accurate enough for standard errors.

One iteration of the full system is

$$\left\{\begin{array}{l} [L_{\beta\beta} - L_{\beta\alpha}(L_{\alpha\alpha})^{-1}L_{\alpha\beta}]\Delta\beta = q = \widetilde{\mathbf{R}}^{\mathbf{T}}[Y - B_{old}\hat{U}_{old}] = L_{\beta} \\ \beta_{new} = \beta_{old} + \Delta\beta \\ \alpha_{new}^{sh} = \log(m_{sh} \,/\, \mathrm{P}_{sh}(\beta_{new}, \hat{U}_{old})), \text{ all } s,h \\ Q_{new} = Q(\alpha_{new}, \beta_{new}) \\ \hat{U}_{new} = \mathbf{1} + (I + D_{old}\, Q_{new})^{-1}\, D_{old}\, B^{\mathbf{T}} A^{-1}(Y - \mu_{new}) \\ \eta_{new} = H(\alpha_{new}, \beta_{new}, \eta_{old}, \hat{U}_{new}) \\ D_{new} = D(\eta_{new}) \end{array}\right\} \qquad (14)$$

Here the notation $Q(\alpha_{new}, \beta_{new})$ means $B^{\mathbf{T}}A^{-1}B$, computed with the new α and β values. Note that the term $B^{\mathbf{T}}A^{-1}\mu$ in the formula for U_{new} is actually the diagonal of Q, written as a vector, so $B^{\mathbf{T}}A^{-1}\mu_{new}$ is available as soon as Q_{new} is. Note also that the matrix $B^{\mathbf{T}}A^{-1}$, although defined in terms of the vector μ, is a constant independent of μ, so that the term $B^{\mathbf{T}}A^{-1}Y$ only needs to be computed once, and indeed

$$\{B^{\mathbf{T}}A^{-1}Y\}^{r} = m^{r}$$

for each leaf-cluster r.

In the actual iterations, the matrix $\hat{K}$ and the right side vector q are computed first, using the previous random effects vector $\hat{U}_{old}$. This is the most time-consuming part of the computation, and it requires special algorithms, discussed in the Appendix, to be done in a reasonable time. The vector μ is never actually formed and stored, although entries are computed, used, and discarded. The updates of $\hat{U}$ and η are done by one of a set of special program modules that implement the covariance models the program recognizes. These modules are self-contained, all with the same interface to the rest of the program, so that it is relatively easy to add new covariance models. The last three lines of (14) are implemented by the code module for the chosen covariance model.

The structure of the iteration (14) can be seen to be Newton-like in the regression coefficients β and Picard-like in the dispersion parameters η. The random effects vector $\hat{U}$ is not an independent solution component, but is determined by β and η, so is just a reporting variable.

It can be shown that the algorithm of (14) amounts to an approximate EM algorithm, the approximation consisting of replacing the conditional expectation of the U's given Y, by the BLUP predictor.

Residuals

Martingale Residuals

The cumulative baseline hazard function $\Lambda_0(t)$ can be estimated Therneau and Grambsch (2000, ch. 4) by

$$\hat{\Lambda}_0(t) = \int_0^t \frac{d\left(\sum_i N_i(s)\right)}{\sum_j \delta_j(s)\exp(\mathbf{R}_j(s)\hat{\beta})}$$

In the notation used here, this is

$$\hat{\Lambda}_0(t) = \sum_{h=1}^{q} I(\tau_h \le t)\frac{m_h}{\sum_{k\in R_{sh}} U^{r^k} w^k \exp(\mathbf{R}^k\beta)}$$

where $I()$ is the indicator function. This simplifies to

$$\begin{aligned}\hat{\Lambda}_0(t) &= \sum_{h=1}^{q} I(\tau_h \le t)\frac{m_h}{\mathrm{P}(\tau_{sh},\beta,\mathbf{U},\mathbf{w})}\\ &= \sum_{\{h|\tau_h \le t\}} \exp(\alpha^h)\end{aligned}$$

since

$$\mathrm{P}_{sh}(\beta,\mathbf{u},\mathbf{w}) = \sum_{k\in R_{sh}} U^{r^k} w^k \exp(\mathbf{R}^k\beta)$$

and

$$\exp(\alpha^{sh}) = m_{sh} \,/\, \mathrm{P}(\tau_{sh},\beta,\mathbf{U},\mathbf{w})$$

The martingale residuals are defined by Therneau and Grambsch as:

$$\hat{M}_i(t) = N_i(t) - \int_0^t \delta_i(t)\exp(\mathbf{R}(t)\hat{\beta})d\hat{\Lambda}_0(t)$$

or in our notation as

$$= N_i(t) - \sum_{\{h|\tau_h \le t \,\&\, i\in R_h\}} U^{r(i)} w^{[i,\tau_h]} \exp(\alpha^h)\exp(\mathbf{R}^i\beta)$$

If subject i has the time interval at risk $(\tau^i_{start}, \tau^i_{end}]$, this is

$$= \begin{cases} 1 - \sum_{\{h|\tau^i_{start} < \tau_h \le \tau^i_{end}\}} U^{r(i)} w^{[i,\tau_h]} \exp(\alpha^h)\exp(\mathbf{R}^i\beta), & t \ge \tau^i_{end} \\ -\sum_{\{h|\tau^i_{start} < \tau_h \le t\}} U^{r(i)} w^{[i,\tau_h]} \exp(\alpha^h)\exp(\mathbf{R}^i\beta), & t \ge \tau^i_{end} \end{cases}$$

$$= \begin{cases} 1 - U^{r(i)}\exp(\mathbf{R}^i\beta)\sum_{\{h|\tau^i_{start} < \tau_h \le \tau^i_{end}\}} w^{[i,\tau_h]}\exp(\alpha^h), & t \ge \tau^i_{end} \\ -U^{r(i)}\exp(\mathbf{R}^i\beta)\sum_{\{h|\tau^i_{start} < \tau_h \le t\}} w^{[i,\tau_h]}\exp(\alpha^h), & t < \tau^i_{end} \end{cases}$$

Or, if the k^{th} data record has interval $(\tau^k_{start}, \tau^k_{end}]$,

$$\hat{M}_k(t) = \begin{cases} 1 - U^{r^k} U^{rk} w^k \exp(\mathbf{R^k}\beta) \sum_{\{h\, k \in R_h\}} \exp(\alpha^h), & t \geq \tau^k_{end} \;\&\; k \in D(\tau^k_{end}) \\ -U^{r^k} w^k \exp(\mathbf{R}^k \beta) \sum_{\{h | k \in R_h\}} \exp(\alpha^h), & t \geq \tau^k_{end} \;\&\; k \in D(\tau^k_{end}) \\ -U^{r^k} w^k \exp(\mathbf{R^k}\beta) \sum_{\{h | \tau^i_{start} < \tau h \leq t\}} \exp(\alpha^h), & t \geq \tau^k_{end} \end{cases}$$

$$= \begin{cases} \chi^k - U^{r^i} w^k \exp(\mathbf{R^k}\beta) \sum_{\{h | k \in R_h\}} \exp(\alpha^h), & t \geq \tau^k_{end} \\ -U^{r^k} w^k \exp(\mathbf{R^k}\beta) \sum_{\{h | \tau^i_{start} < \tau h \leq t\}} \exp(\alpha^h), & t < \tau^k_{end} \end{cases}$$

and

$$\int_0^\infty f(t) d\hat{M}_k(t) = \sum_{h=1}^{q} f(\tau_h)[\hat{M}_k(\tau_h) - \hat{M}_k(\tau_{h-1}]$$
$$= f(\tau^k_{end})\chi^k - \sum_{\{h | k \in R_h\}} f(\tau_h) U^{r^k} w^k \exp(\alpha^h) \exp(\mathbf{R}^k \beta)$$

Note that

$$\{h | k \in \mathrm{R}_h\} = \{h | \tau_h \in (\tau^k_{start}, \tau^k_{end}]\}$$

Score Residuals

Therneau and Grambsch (2000) define the score residuals as follows: first let

$$\bigcup_i (\gamma, t) = \int_0^t [\mathbf{R^i}(t) - \bar{r}(\beta, s)] d\hat{M}_i(s)$$

where

$$\bar{r}(\beta,t) = \frac{\sum_j \delta_j(t)\exp(\mathbf{R}^j(t)\beta)\mathbf{R}^j(t)}{\sum_j \delta_j(t)\exp(\mathbf{R}^j(t)\beta)}$$

or

$$\bar{r}(\beta,t) = \frac{\sum_{k\in R(t)} U^{r^k} w^k \exp(\mathbf{R}^k\beta)\mathbf{R}^k}{\sum_{k\in R(t)} U^{r^k} w^k \exp(\mathbf{R}^k\beta)}$$

$$\bar{r}(\beta,t) = \frac{\sum_{k\in R(t)} U^{r^k} w^k \exp(\mathbf{R}^k\beta)\mathbf{R}^k}{\mathbb{P}(t,\beta,\mathbf{U},\mathbf{w})}$$

Now: the score residual $\bigcup_i(\gamma)$ is defined as

$$\bigcup_i(\gamma) = \bigcup_i(\gamma,\infty) = \int_0^\infty [\mathbf{R}^{\mathbf{i}} - \bar{r}(\beta,s)]d\hat{M}_i(s)$$

or

$$\bigcup_i(\gamma) = \sum_{h=1}^{q}[\mathbf{R}^k - \bar{r}(\beta,\tau_h)]\hat{M}_k(\tau_h) - \hat{M}_k(\tau_h - 1)]$$

$$= [\mathbf{R}^k - \bar{r}(\beta,\tau^k_{end})]\chi^k \sum_{\{h|k\in R_h\}} - U^{r^k} w^k \exp(\alpha^h)\exp(\mathbf{R}^k\beta)\mathbf{R}^k - \bar{r}(\beta,\tau_h)]$$

$$= [\mathbf{R}^k - \bar{r}(\beta,\tau^k_{end})]\chi^k - U^{r^k} w^k \exp(\mathbf{R}^k\beta) \sum_{\{h|k\in R_h\}} \exp(\alpha^h)\mathbf{R}^k - \bar{r}(\beta,\tau_h)$$

or,

$$\bigcup_k(\gamma) = [\mathbf{R}^k - \bar{r}(\beta,\tau^k_{end})]\chi^k - U^{r^k} w^k \exp(\mathbf{R}^k\beta) \sum_{\{h|k\in R_h\}} \exp(\alpha^h)[\mathbf{R}^k - \bar{r}(\beta,\tau_h)]$$

$$= [\mathbf{R}^k - \bar{r}(\beta,\tau^k_{end})]\chi^k - U^{r^k} w^k \exp(\mathbf{R}^k\beta)\mathbf{R}^k \sum_{\{h|k\in R_h\}} \exp(\alpha^h)$$

$$+ U^{r^k} w^k \exp(\mathbf{R}^k\beta) \sum_{\{h|k\in R_h\}} \exp(\alpha^h)\bar{r}(\beta,\tau_h)$$

Now

$$\sum_k \cup^k (\gamma) =$$

$$\sum_k \left[[\mathbf{R}^k - \bar{r}(\beta, \tau^k_{end})]\chi^k - U^{r^k} w^k \exp(\mathbf{R}^k \beta) \sum_{\{h | k \in R_h\}} \exp(\alpha^h)[\mathbf{R}^k - \bar{r}(\beta, \tau_h)] \right]$$

$$= \sum_k [\mathbf{R}^k - \bar{r}(\beta, \tau^k_{end})]\chi^k - \sum_k \left[U^{r^k} w^k \exp(\mathbf{R}^k \beta) \sum_{\{h | k \in R_h\}} \exp(\alpha^h)[\mathbf{R}^k - \bar{r}(\beta, \tau_h)] \right]$$

$$= \sum_k \mathbf{R}^k \chi^k - \sum_{h=1}^{q} \exp(\alpha^h) \sum_{k \in R_h} U^{r^k} w^k \exp(\mathbf{R}^k \beta)\mathbf{R}^k +$$

$$\sum_{h=1}^{q} \exp(\alpha^h)\bar{r}(\beta, \tau_h) \sum_{k \in R_h} U^{r^k} w^k \exp(\mathbf{R}^k \beta) - \sum_k \bar{r}(\beta, \tau^k_{end})\chi^k$$

$$= \sum_{h=1}^{q} \left[\sum_{k \in R_h} \mathbf{R}^k \chi^k - \exp(\alpha^h) \sum_{k \in R_h} U^{r^k} w^k \exp(\mathbf{R}^k \beta)\mathbf{R}^k \right] +$$

$$\sum_{h=1}^{q} \exp(\alpha^h)\bar{r}(\beta, \tau_h) \sum_{k \in R_h} U^{r^k} w^k \exp(\mathbf{R}^k \beta) - \sum_k \bar{r}(\beta, \tau^k_{end})\chi^k$$

The third and fourth terms are

$$\sum_{h=1}^{q} \exp(\alpha^h)\bar{r}(\beta, \tau_h) \sum_{k \in R_h} U^{r^k} w^k \exp(\mathbf{R}^k \beta) - \sum_k \bar{r}(\beta, \tau^k_{end})\chi^k$$

$$= \sum_{h=1}^{q} \frac{m_h}{\mathbb{P}_h(\beta, \mathbf{U}, \mathbf{w})} \bar{r}(\beta, \tau_h)\mathbb{P}_h(\beta, \mathbf{U}, \mathbf{w}) - \sum_{h=1}^{q} \sum_{k \in D_h} \bar{r}(\beta, \tau^k_{end})$$

$$= \sum_{h=1}^{q} m_h \bar{r}(\beta, \tau_h) - \sum_{h=1}^{q} \sum_{k \in D_h} \bar{r}(\beta, \tau_h)$$

$$= \sum_{h=1}^{q} m_h \bar{r}(\beta, \tau_h) - \sum_{h=1}^{q} m_h \bar{r}(\beta, \tau_h) = 0$$

so, we have

$$\sum_k \cup_k (\beta) = L_\beta$$

and

$$\bar{r}(\beta, \tau_{end}^k) = \frac{\sum_{j \in R(\tau_{end}^k)} U^{r^j} w^j \exp(\mathbf{R}^j \beta) \mathbf{R}^j}{\sum_{j \in R(\tau_{end}^k)} U^{r^j} w^j \exp(\mathbf{R}^j \beta)}$$

$$\bar{r}(\beta, \tau_h) = \frac{\sum_{j \in R_h} U^{r^j} w^j \exp(\mathbf{R}^j \beta) \mathbf{R}^j}{\sum_{j \in R_h} U^{r^j} w^j \exp(\mathbf{R}^j \beta)}$$
$$= \frac{\exp(\alpha^h)}{m_h} \sum_{j \in R_h} U^{r^j} w^j \exp(\mathbf{R}^j \beta) \mathbf{R}^j$$

Note that the denominator of $\bar{r}(\beta, t)$ is

$$\mathtt{P}(t, \beta, \mathbf{U}, \mathbf{w}) = \sum_{k \in R(t)} U^{r^k} w^k \exp(\mathbf{R}^k \beta)$$

and

$$\bar{r}(\beta, t) = \frac{\partial}{\partial \beta} \log(\mathtt{P}(t, \beta, \mathbf{u}, \mathbf{w}))$$
$$=$$

The score residuals form a $N_M \times p$ matrix $\mathtt{U}(\gamma)$. Note also that at the solution we have

$$\sum_k \bigcup_k (\beta) = L_\beta = 0$$

***dfbeta* Residuals**

The *dfbeta* residuals Therneau and Grambsch (2000, ch. 7.1) form a $N_M \times p$ matrix $\mathtt{D}(\gamma)$, defined from the score residual matrix $\bigcup(\gamma)$ as

$$\mathtt{D}(\gamma) = \bigcup(\gamma) K(\gamma)^{-1}$$

where $K(\gamma)$ is the Schur-complement matrix used for standard errors and defined in (27). The *dfbeta* residuals are used by the program to compute robust variance estimates.

Models for var(U)

We give a few standard forms for $D = \text{var}(U)$. Each of the forms contains undetermined parameters, called "dispersion parameters". These must be estimated, generally by estimating functions involving $\hat{U}$.

One-Level Distance-Decay Form

General

Let d_{rs} denote the distance between clusters r and s, in a one-level hierarchy. The distance can be any suitable distance measure, satisfying

$$\begin{aligned} & d_{rr} = 0 \\ & d_{rs} > 0 \text{ if } r \neq s \\ & d_{rs} = d_{sr} \\ & \quad \text{and sometimes} \\ & d_{rs} \leq d_{rq} + d_{qs} \end{aligned}$$

The last is the triangle inequality, which may not be satisfied by some distance measures.

Suppose that $\mathbf{w}$ is a fixed vector of cluster weights (e.g. city populations, etc.), which is supplied as data. It is assumed that $\mathbf{w}_r > 0$, for all r. We specify

$$D_s^r = \text{cov}(U_r, U_s) = \sigma^2 \, \mathbf{w}_r \, \mathbf{w}_s \; g(d_{rs} / h)$$

where $g = g(x, \cdots)$ is a function of $x = d / h$, with possibly other parameters. We require

$$\left\{\begin{array}{l} g(0)=1 \\ g \text{ continuous} \\ g \text{ decreasing} \\ g(x)\geq 0 \\ g(x)\to 0 \text{ as } x\to\infty \\ \text{and sometimes} \\ g(x)=0 \text{ if } x\geq 1 \end{array}\right\} \tag{15}$$

The first requirement is just a normalization. The last requirement, which may or may not be invoked in a particular problem, gives some sparseness to the matrix D. The choice of $x=1$ as the cutoff value is arbitrary, since it is just a matter of the scaling of h.

More generally, we note that h is not really a parameter of the distribution, but simply a normalization of the distance function $d()$. The function $d()$ and the normalizing constant h are assumed to be chosen a priori. The choice of h reflects the spatial resolution that is thought to be important.

We also require that the matrix D be positive definite, which restricts the choice of functions g and parameter values h. some possibilities are:

$$\begin{aligned} g(x) &= \exp(-x) \\ g(x) &= \exp(-x^2) \\ g(x) &= (1-x)^2 \\ g(x) &= 1-x \\ g(x) &= \rho^x; \quad 0<\rho<1 \end{aligned}$$

This last is the same as

$$g(x)=\exp(\log(\rho)\,x)$$

So the parameter ρ is essentially equivalent to the resolution parameter h. Now let $d_0 = \min_{r\neq s}(d_{rs})$. If h is chosen small enough that $0<g(d_0/h)<1/N_{leaf}$, where N_{leaf} is the number of clusters, then D is positive definite:

$$\sum_{s\neq r} g(d_{rs}/h) < \sum_{s\neq r} g(d_0/h) < (N_{leaf}-1)/N_{leaf} < 1 = g(d_{rr}/h)$$

so the matrix without the weights is diagonally dominant, and hence positive definite. The weighted matrix is just a matter of multiplying on left and right by a positive diagonal matrix, which preserves positive definiteness. We conclude that, given the metric $\{d_{rs}\}$, and given an arbitrary function g having the first five properties specified in (15), there is a non-empty open interval $(0, h_m)$ of h-values that make D positive definite. This does not require the distance to satisfy the triangle inequality.

Example: ρ^x

Now let us consider the last example listed above in more detail. We have $g(x) = \rho^x$, for some ρ with $0 < \rho < 1$, and

$$D_s^r = \mathrm{cov}(U_r, U_s) = \sigma^2 \mathbf{w}_r \mathbf{w}_s \rho^{d_{rs}/h} = \sigma^2 \mathbf{w}_r \mathbf{w}_s \exp(\log(\rho) d_{rs}/h)$$

As already pointed out, the parameters ρ and h are redundant, since whatever value is chosen for h, we can replace it with $\tilde{h} = -h/\log(\rho)$, and absorb ρ into $\tilde{h}$, giving

$$D_s^r = \mathrm{cov}(U_r, U_s) = \sigma^2 \mathbf{w}_r \mathbf{w}_s \exp(-d_{rs}/\tilde{h})$$

Or, we can absorb h into $\tilde{\rho}$, defining

$$\begin{aligned} \log(\tilde{\rho}) &= \log(\rho)/h \\ \tilde{\rho} &= \exp(\log(\rho)/h) \\ &= \rho^{1/h} \end{aligned}$$

So, without loss of generality we can take either $h = 1$, and consider

$$\mathrm{cov}(U_r, U_s) = \sigma^2 \mathbf{w}_r \mathbf{w}_s \rho^{d_{rs}}$$

or take $\rho = e^{-1}$ and consider

$$\mathrm{cov}(U_r, U_s) = \sigma^2 \mathbf{w}_r \mathbf{w}_s \exp(-d_{rs}/h)$$

These are equivalent. Let us choose the first form, and consider

$$D_s^r = \mathrm{cov}(U_r, U_s) = \sigma^2 \mathbf{w}_r \mathbf{w}_s \rho^{d_{rs}}$$

As mentioned, the parameter ρ is just a matter of the scaling of the metric d_{rs}. However, let us suppose the metric is already fixed: in that case, ρ becomes a parameter to be fitted from the data.

Estimating σ^2 and ρ

We have seen from (10)

$$D = \mathrm{var}(U) = \mathrm{var}(\hat{U}) + (I + DQ)^{-1} D$$

and

$$D_r^r = \varepsilon((\hat{U}^r - 1)^2) + \{(I + DQ)^{-1} D\}_r^r$$
$$D_s^r = \varepsilon((\hat{U}^r - 1)(\hat{U}^s - 1)) + \{(I + DQ)^{-1} D\}_s^r$$

Note that

$$(I + DQ)^{-1} D = (D^{-1} + Q)^{-1}$$

is positive definite, so

$$\{(I + DQ)^{-1} D\}_r^r > 0$$

We note that under the assumed form of D,

$$D_r^r = \mathbf{w}_r^2 \sigma^2, \text{ all } r$$

This leads to the equation

$$\sigma^2 = \mathbf{w}_r^{-2} \left[\varepsilon((\hat{U}^r - 1)^2) + \{(I + DQ)^{-1} D\}_r^r \right]$$

for all r, and estimating the expectation by a sample average, we have the estimating equation

$$\sigma^2 = \underset{r}{\text{average}}(\mathbf{w}_r^{-2}[(\hat{U}^r - 1)^2 + \{(I + DQ)^{-1} D\}_r^r])$$

where the average is taken over all leaf clusters r. Note that the right side depends on D, which depends on σ^2. We can regard this as an equation in σ^2, which occurs on both sides. Its solution $\hat{\sigma}^2$ is necessarily positive.

However, the above procedure for $\hat{\sigma}^2$ may not be very robust when the weight vector has some small values, since those terms will tend to dominate the estimate. We proceed instead as follows: define

$$K_r^r = \left[(\hat{U}^r - 1)^2 + \{(I + DQ)^{-1} D\}_r^r\right]$$

and choose $\hat{\sigma}^2$ to satisfy

$$\min_{\sigma^2}\{e_0(\sigma^2) | \sigma^2 > 0\} = \min_{\sigma^2}\left\{\sum_r [K_r^r - \sigma^2 \mathbf{w}_r^2]^2 \;\middle|\; \sigma^2 > 0\right\}$$

This gives

$$\partial e_0 / \partial \sigma^2 = -2 \sum_r [K_r^r - \sigma^2 \mathbf{w}_r^2] \mathbf{w}_r^2 = 0$$

which leads to

$$\hat{\sigma}^2 = \frac{\sum_r \mathbf{w}_r^2 K_r^r}{\sum_r \mathbf{w}_r^4}$$

which should be less sensitive to small weight values. Again, the right side depends on D, which depends on σ^2.

Now consider ρ; we have

$$D_s^r = \sigma^2 \mathbf{w}_r \mathbf{w}_s \rho^{d_{rs}} = \varepsilon((\hat{U}^r - 1)(\hat{U}^s - 1)) + \{(I + DQ)^{-1} D\}_s^r$$

For some "contiguity" metrics, we can do something for ρ similar to the above method for σ^2. However, it requires that there be many pairs with $d_{rs} = 1$, and this is a strong assumption on the metric. Instead we can proceed as follows: let

$$K_s^r = \left[(\hat{U}^r - 1)(\hat{U}^s - 1) + \{(I + DQ)^{-1} D\}_s^r\right]$$

or

$$K = \left[(\hat{U} - \mathbf{1})(\hat{U} - \mathbf{1})^{\mathbf{T}} + (I + DQ)^{-1} D\right]$$

Or, if D is large and sparse, we might want to approximate K by

$$K \cong \left[(\hat{U} - \mathbf{1})(\hat{U} - \mathbf{1})^{\mathbf{T}} + (I + \text{diag}(DQ))^{-1} D\right]$$
$$K \cong \left[(\hat{U} - \mathbf{1})(\hat{U} - \mathbf{1})^{\mathbf{T}} + Q^{-1} \text{diag}(Q^{-1} + D)^{-1} D\right]$$

or some similar sparse approximation. The reason for this is that sparse matrices rarely have sparse inverses, and forming $(I + DQ)^{-1}$ may not be feasible.

Then we want to choose σ^2 and either ρ or h so that the matrix $\{\sigma^2 \mathbf{w}_r \mathbf{w}_s \rho^{d_{rs}}\}$ or $\{\sigma^2 \mathbf{w}_r \mathbf{w}_s \exp(-d_{rs}/h)\}$ most closely resembles K. We must have $0 < \rho < 1$, or $h > 0$. We can try minimizing the difference in some matrix norm, such as the Frobenius norm, separating the diagonal and off-diagonal entries:

$$\min_{\sigma^2} \left\{ \sum_r [K_r^r - \sigma^2 \mathbf{w}_r^2]^2 \;\middle|\; \sigma^2 > 0 \right\}$$

and

$$\min_h \left\{ \sum_{r \neq s} [K_s^r - \sigma^2 \mathbf{w}_r \mathbf{w}_s \rho^{d_{rs}}]^2 \;\middle|\; h > 0 \right\}$$

where σ^2 is taken as given, in the second line. Then $\hat{\sigma}^2$ is determined as before. Let

$$e(\rho) = \sum_{r \neq s} \sum_{r \neq s} [K_s^r - \sigma^2 \mathbf{w}_r \mathbf{w}_s \rho^{d_{rs}})]^2$$

We want to minimize this for $0 \le \rho \le 1$. There are many codes available for one-dimensional minimization problems; the program uses one called fmin, due originally to Richard Brent.

Another Approach to Estimating ρ

Suppose that d_{rs} is a neighbor-type distance matrix, i.e. every value is either 1 or ∞, meaning that clusters are either neighbors or strangers. Assume for the moment that we have an estimate $\hat{\sigma}^2$ for σ^2, from somewhere. We have from (9) that if κ and ν are *Z*-indices, with $\kappa \neq \nu$, then

$$\operatorname{cov}(Y^\kappa, Y^\nu) = \mu^\kappa \mu^\nu D^{r^\kappa}_{r^\nu}$$

where D, again, is $\operatorname{cov}(U,U)$, and r^κ is the leaf-cluster corresponding to the *Z*-index κ. Now let r,s be leaf clusters. We write $r \leftrightarrow s$ to mean $d_{rs} = 1$, i.e. r and s are neighbors. Then

$$D^r_s = \frac{\operatorname{cov}(Y^\kappa, Y^\nu)}{\mu^\kappa \mu^\nu} = \operatorname{cov}\left(\frac{Y^\kappa}{\mu^\kappa}, \frac{Y^\nu}{\mu^\nu}\right)$$

for any $\kappa = [e,\tau]$ and $\nu = [g,\xi]$ with $e \in r$ and $g \in s$. We will abuse notation by writing $\kappa \in r$ and $\nu \in s$. It follows that

$$D^r_s = \operatorname{mean}\left\{\operatorname{cov}\left(\tfrac{Y^\kappa}{\mu^\kappa}, \tfrac{Y^\nu}{\mu^\nu}\right) \,\middle|\, \forall \kappa \in r,\ \forall \nu \in s\right\} \tag{16}$$

Noting that

$$\varepsilon\left(\tfrac{Y^\kappa}{\mu^\kappa}\right) = 1$$

we can estimate the right side of (16) by

$$\begin{aligned}\hat{D}_s^r &= \text{mean}\left\{\left(\frac{Y^\kappa}{\mu^\kappa}-1\right)\left(\frac{Y^\nu}{\mu^\nu}-1\right) \,\middle|\, \forall \kappa \in r,\ \forall \nu \in s\right\} \\ &= \frac{1}{n_r n_s}\sum_{\kappa \in r}\sum_{\nu \in s}\left(\frac{Y^\kappa}{\mu^\kappa}-1\right)\left(\frac{Y^\nu}{\mu^\nu}-1\right) \\ &= \underset{\kappa \in r}{\text{mean}}\left(\frac{Y^\kappa}{\mu^\kappa}-1\right)\underset{\nu \in s}{\text{mean}}\left(\frac{Y^\nu}{\mu^\nu}-1\right) \\ &= M^r M^s\end{aligned}$$

say, where we define

$$M^r = \underset{\kappa \in r}{\text{mean}}\left(\frac{Y^\kappa}{\mu^\kappa}-1\right)$$

which is easily computed for all r on each iteration. Now, we are postulating

$$D_s^r = \sigma^2 \rho^{d_{rs}} = \begin{cases}\sigma^2\rho & \text{if } r \leftrightarrow s \\ 0 & \text{else}\end{cases}$$

It follows that

$$\sigma^2 \rho \approx \underset{r \leftrightarrow s}{\text{mean}}(M^r M^s)$$

where $\underset{r \leftrightarrow s}{\text{mean}}$ denotes the mean over all pairs (r,s) that are neighbors, and so we define the estimator

$$\hat{\rho} = \frac{\underset{r \leftrightarrow s}{\text{mean}}(M^r M^s)}{\hat{\sigma}^2}$$

One-Level Moving Average

Suppose that d_{rs} is a distance matrix; we derive a matrix A from d as follows: replace any infinite entries by 0, and scale the rows so that each row sums to 1. The matrix A then has the properties:

- $A \geq 0$, and $A_{rs} > 0$ if clusters r and s are neighbors.
- $A\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the vector of all 1's.

- $\mathrm{diag}(A)$ is 0.
- ΔA is symmetric, for some diagonal matrix Δ.

We postulate the following model for the random effects vector $U = \{U_r\}$:

$$U = (1-\rho)V + \rho AV = P(\rho)V$$

where:

- V is a random vector assumed iid and $V > 0$, $\varepsilon(V) = \mathbf{1}$, and $\mathrm{var}(V) = \sigma^2 I$
- σ^2 and ρ are parameters to be estimated, with $\sigma^2 > 0$ and $0 \le \rho < \frac{1}{2}$

then, letting $P(\rho)$ be

$$P(\rho) = (1-\rho)I + \rho A$$

we have $\varepsilon(U)=1$, and the covariance matrix of U is

$$\begin{aligned}\mathrm{var}(U) = D(\sigma^2,\rho) &= \sigma^2 P(\rho)P(\rho)^{\mathbf{T}} \\ &= \sigma^2\left[(1-\rho)^2 I + \rho(1-\rho)(A + A^{\mathbf{T}}) + \rho^2 AA^{\mathbf{T}}\right]\end{aligned}$$

Still supposing that the distance matrix is neighbor type, we have as above, for $r \ne s$,

$$\underset{r\leftrightarrow s}{\mathrm{mean}}(D_s^r) \approx \underset{r\leftrightarrow s}{\mathrm{mean}}(M^r M^s)$$

or

$$\sigma^2[\rho(1-\rho)\underset{r\leftrightarrow s}{\mathrm{mean}}(A + A^{\mathbf{T}}) + \rho^2 \underset{r\leftrightarrow s}{\mathrm{mean}}(AA^{\mathbf{T}})] \approx \underset{r\leftrightarrow s}{\mathrm{mean}}(M^r M^s)$$

The left side is

$$\begin{aligned}&\sigma^2[2\rho(1-\rho)\underset{r\leftrightarrow s}{\mathrm{mean}}(A_s^r) + \rho^2 \underset{r\leftrightarrow s}{\mathrm{mean}}(\sum_t A_t^r A_t^s)] \\ &= \sigma^2[2\rho(1-\rho)R + \rho^2 S]\end{aligned}$$

where R and S need to be computed from A only once. So we have the quadratic equation in ρ:

$$2\rho(1-\rho)R + \rho^2 S = \frac{\underset{r \leftrightarrow s}{\text{mean}}(M^r M^s)}{\hat{\sigma}^2}$$

The left side is 0 when $\rho = 0$. If $S < 2R$, then the left side is increasing in ρ on $[0, \frac{1}{2})$, so ρ is either uniquely defined, or must be taken to be $\frac{1}{2}$ (i.e. the model doesn't fit well). If $S > 2R$, then there is no positive solution, so we must take $\rho = 0$, again observing that the model doesn't fit well.

Multi-Level Nested Form

Definitions and Properties

Assumptions:

Suppose we have a nested system (tree) of clusters (sets of individuals), ordered by the "$\subseteq$" relation. Let the "level-1" clusters be those clusters with no parent (no larger cluster of which the given one is a subset). Let the "leaves", or "finest-level" clusters be those with no children. A cluster can be both root and leaf. We define the level of a cluster c recursively by:

- The level of a level-1 cluster is 1
- If the parent of c has level ℓ, then c has level $\ell + 1$.

The clusters of any level ℓ are disjoint. Let N_{leaf} be the total number of leaves, ordered lexically (this assumes a multi-index representation of the cluster tree). Recall that U is a leaf-vector, so has dimension N_{leaf}. Let L be the highest level in the tree.

We are now going to assign a random effect vector U_i to every cluster i at any level, not merely the leaves. If i is a cluster at some level, let $\mathtt{P}(i)$ be the set consisting of i,

the parent of i, the grandparent of i,…, back to level 1. We also write $U\{P(i)\}$ to denote the vector of values of all the corresponding random effects of i and its ancestors. We will write $U\{\ell\}$ to denote the vector of level-ℓ random effects. So what we have been denoting by U so far is $U\{L\}$. We write $U\{\ell\}^i$ to denote the i^{th} component of the vector $U\{\ell\}$. We will suppress the "$\{\ell\}$" when it is clear from the context.

We will impose the following assumptions:

1) If i is a cluster of level ℓ, and $\{i1,i2,\cdots,iq\}$ are the children of i, then

$U^{i1},U^{i2},\ldots,U^{iq}$ are conditionally IID, given $U\{P(i)\}$, and

$$\varepsilon(U_{ij}|U\{P(i)\}) = U^i$$
$$\text{var}(U_{ij}|U\{P(i)\}) = \sigma^2_{\ell+1}U^i$$

where $\sigma^2_{\ell+1}$ is a dispersion parameter that must be estimated.

2) If r is a leaf-cluster, and $e_1, e_2, \ldots, e_p$ are the individuals in r, and

$\kappa_1 = [e_1, \tau_1], \kappa_2 = [e_2, \tau_2], \ldots, \kappa_p = [e_p, \tau_p]$ are pairs in Z, then the Y^{κ_j} are conditionally independent given $U\{P(r)\}$, and

$$\varepsilon(Y^{\kappa_j}|U\{P(r)\}) = \mu^{\kappa_j}U^r$$

To make these statements apply to level 1, we define a "root" or level-0 cluster to be the set of all individuals, and then each level-1 cluster is a child of the root. Define the root's random effect value (degenerate) as $U\{0\} = 1$. Then the above assumptions also hold for level-1 clusters.

Definition: Sometimes we will consider the cluster tree truncated or trimmed at level $\ell < L$. By the ℓ-trimmed tree, we mean the tree with all clusters of higher level than ℓ

removed, so that the leaves of the trimmed tree (the "ℓ-leaves") are the clusters of level ℓ, together with the leaves of the full tree that are of level less than ℓ.

For a 1-level model, the assumptions clearly imply $\text{cov}(U,U) = \sigma_1^2 I$. For a 2-level model, two level-2 clusters are independent if they are children of different parents. If clusters ij and ik are both children of cluster i, with $j \neq k$, then

$$\begin{aligned}\text{cov}(U^{ij},U^{ik}) &= \varepsilon_i(\text{cov}(U^{ij},U^{ik}|U^i)) + \text{cov}_i(\varepsilon(U^{ij}|U^i),\varepsilon(U^{ik}|U_i)) \\ &= 0 + \text{cov}_i(U^i,U^i) = \sigma_1^2\end{aligned}$$

and

$$\begin{aligned}\text{var}(U^{ij}) &= \varepsilon_i(\text{var}(U^{ij}|U^i)) + \text{var}_i(\varepsilon(U^{ij}|U^i)) \\ &= \varepsilon_i(\sigma_2^2 U^i) + \text{var}_i(U^i) = \sigma_2^2 + \sigma_1^2\end{aligned}$$

So if all leaves are at level 2, we have

$$D = \text{cov}(U,U) = \sigma_1^2 M + \sigma_2^2 I$$

where M is block-diagonal, with a block for each main (i.e. level-1) cluster, and each block M_i is all 1's on the rows and columns corresponding to cluster i.

Continuing this sequence, consider 3-level models: suppose that cluster i is at level 2, that i's children are $\{i1,i2,\cdots,iq\}$, and that i's parent is c. Then if $j \neq k$,

$$
\begin{aligned}
\operatorname{cov}(U^{ij},U^{ik}) &= \varepsilon_{i,c}(\operatorname{cov}(U^{ij},U^{ik}|U^{i},U^{c})) + \\
&\qquad \operatorname{cov}_{i,c}(\varepsilon(U^{ij}|U^{i},U^{c}),\varepsilon(U^{ik}|U^{i},U^{c})) \\
&= 0 + \operatorname{cov}_{i,c}(U^{i},U^{i}) = \operatorname{var}(U^{i}) \\
&= \varepsilon_{c}(\operatorname{var}(U^{i}|U^{c})) + \operatorname{var}_{c}(\varepsilon(U^{i}|U^{c})) \\
&= \varepsilon_{c}(\sigma_{2}^{2}U^{c}) + \operatorname{var}_{c}(U^{c}) = \sigma_{2}^{2} + \sigma_{1}^{2} \\
&\text{and} \\
\operatorname{var}(U^{ij}) &= \varepsilon_{i,c}(\operatorname{var}(U^{ij}|U^{i},U^{c})) + \operatorname{var}_{i,c}(\varepsilon(U^{ij}|U^{i},U^{c})) \\
&= \varepsilon(\sigma_{3}^{2}U^{i}) + \operatorname{var}(U^{i}) \\
&= \sigma_{3}^{2} + \sigma_{2}^{2} + \sigma_{1}^{2}
\end{aligned}
$$

It follows that if all leaves are at level 3, we have

$$D = \operatorname{cov}(U,U) = \sigma_{1}^{2}\,M + \sigma_{2}^{2}L + \sigma_{3}^{2}\,I$$

where M is block-diagonal with blocks M_i corresponding to level-1 clusters, and each M_i is all 1's on the leaves descending from cluster i. Similarly, the matrix L is block-diagonal with blocks L_j corresponding to level-2 clusters, and each L_j is all 1's on the leaves that are children of j.

We also have: if cluster i is at level 1, and i's children are {*i*l, *i*2, , *iq*}, , at level 2, and if also cluster m is at level 1, then

$$
\begin{aligned}
\operatorname{cov}(U^{i},U^{m}) &= \sigma_{1}^{2}\,\delta_{im} \\
\operatorname{cov}(U^{ij},U^{ik}) &= \varepsilon_{i}(\operatorname{cov}(U^{ij},U^{ik}|U^{i})) + \operatorname{cov}_{i}(\varepsilon(U^{ij}|U^{i}),\varepsilon(U^{ik}|U^{i})) \\
&= 0 + \operatorname{cov}(U^{i},U^{i}) = \sigma_{1}^{2} \\
\operatorname{var}(U^{ij}) &= \varepsilon_{i}(\operatorname{var}(U^{ij}|U^{i})) + \operatorname{var}_{i}(\varepsilon(U^{ij}|U^{i})) \\
&= \varepsilon_{i}(\sigma_{2}^{2}U^{i}) + \operatorname{var}_{i}(U^{i}) = \sigma_{2}^{2} + \sigma_{1}^{2}
\end{aligned}
$$

It follows that

$$
\begin{aligned}
D\{1\} &= \mathrm{cov}(U\{1\}, U\{1\}) = \sigma_1^2 I^1 \\
D\{2\} &= \mathrm{cov}(U\{2\}, U\{2\}) = \sigma_1^2 M_1^2 + \sigma_2^2 I^2
\end{aligned}
\tag{17}
$$

where M_1^2 is block-diagonal with blocks of all 1's corresponding to the level-1 clusters. Note that $\mathrm{cov}(U\{1\}, U\{1\})$ is $n_1 \times n_1$, where n_1 is the number of level-1 clusters, and $\mathrm{cov}(U\{2\}, U\{2\})$ is $n_2 \times n_2$, with n_2 the number of level-2 clusters. The notations I^1 and I^2 mean the $n_1 \times n_1$ and $n_2 \times n_2$ identities, respectively. In this notation scheme, we can write the leaf-level covariance above as

$$
D^3 = \mathrm{cov}(U^3, U^3) = \sigma_1^2 M_1^3 + \sigma_2^2 M_2^3 + \sigma_3^2 I^3
$$

so M_1^3 corresponds to M, M_2^3 to L.

Suppose now that i and m are at level ℓ, with highest-level common ancestor π. Then it can be shown that U^i and U^m are conditionally independent given U^π, and we have, if $i \neq m$,

$$
\begin{aligned}
\mathrm{cov}(U^i, U^m) &= \varepsilon(\mathrm{cov}(U^i, U^m | U^\pi)) + \mathrm{cov}(\varepsilon(U^i | U^\pi), \varepsilon(U^m | U^\pi)) \\
&= 0 + \mathrm{var}(U^\pi)
\end{aligned}
\tag{18}
$$

and if $i = m$, then π is the parent of i:

$$
\begin{aligned}
\mathrm{cov}(U^i, U^i) &= \varepsilon(\mathrm{cov}(U^i, U^i | U^\pi)) + \mathrm{cov}(\varepsilon(U^i | U^\pi), \varepsilon(U^i | U^\pi)) \\
&= \varepsilon(\sigma_\ell^2 U^\pi) + \mathrm{var}(U^\pi) \\
&= \sigma_\ell^2 + \mathrm{var}(U^\pi),
\end{aligned}
\tag{19}
$$

say. We can therefore proceed by induction on the level ℓ, to derive expressions for the covariance matrix of the U's at any level, analogous to (17). The general form is given in (22).

Examples

We now consider two examples.

Example 1: Consider the following simple cluster hierarchy. There are two clusters: Cluster 1 has subclusters 1.1 and 1.2, and Cluster 2 is not subdivided into subclusters. Then the U -vector consists of $[U^{1.1},\ U^{1.2},\ U^{2}]^{\mathbf{T}}$, corresponding to the leaves, and we have

$$\text{var}(Y) = A + \sigma^2[\mu^{[1]}(\mu^{[1]})^{\mathbf{T}} + \mu^{[2]}(\mu^{[2]})^{\mathbf{T}}] + \omega^2[\mu^{[1.1]}(\mu^{[1.1]})^{\mathbf{T}} + \mu^{[1.2]}(\mu^{[1.2]})^{\mathbf{T}}]$$

where we let $\sigma^2 = \sigma_1^2$ and $\omega^2 = \sigma_2^2$. Then,

$$D = D^2 = \text{cov}(U,U) = \sigma^2\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \omega^2\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Example 2: Consider a three-level example: clusters 1 and 2 are the same as in Example 1. Cluster 3 has subclusters 3.1 and 3.2, and subcluster 3.1 has subsubclusters 3.1.1 and 3.1.2. Then

$$U = [U_{1.1},\ U_{1.2},\ U_2,\ U^{3.1.1},\ U^{3.1.2},\ U^{3.2}]^{\mathbf{T}}$$

or (in another notation)

$$U = [U^{1.1.0},\ U^{1.2.0},\ U^{2.2.0},\ U^{3.1.1},\ U^{3.1.2},\ U^{3.2.0}]^{\mathbf{T}}$$

Then

$$\operatorname{cov}(U,U) = \sigma^2 \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} + \omega^2 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$+ \xi^2 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

General Forms For Covariance: Leaf Clusters

Now we need a few definitions: For each leaf r, let ζ_r be the indicator leaf-vector of leaf r, i.e.

$$(\zeta_r)^j = \delta_{rj}$$

The right side is the Kronecker delta. For each non-leaf cluster i, let ζ_i be the indicator leaf-vector of i, i.e. $\zeta_i = 1$ on the leaves that descend from i, and 0 otherwise. Then clearly

$$\zeta_i = \sum_{r \subseteq i} \zeta_r$$

where $r \subseteq i$ means that leaf-cluster r is a descendant of cluster i. Let L be the maximum level in the cluster-tree.

For Example 2 above, we have $L = 3$, and for the leaves,

$$
\begin{aligned}
\zeta_{1.1} &= [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^{\mathbf{T}} \\
\zeta_{1.2} &= [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]^{\mathbf{T}} \\
\zeta_{2} &= [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^{\mathbf{T}} \\
\zeta_{3.1.1} &= [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]^{\mathbf{T}} \\
\zeta_{3.1.2} &= [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0]^{\mathbf{T}} \\
\zeta_{3.2} &= [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]^{\mathbf{T}}
\end{aligned}
$$

For the coarser clusters, we have

$$
\begin{aligned}
\zeta_{1} &= [1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]^{\mathbf{T}} \\
\zeta_{3.1} &= [0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0]^{\mathbf{T}} \\
\zeta_{3} &= [0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1]^{\mathbf{T}}
\end{aligned}
$$

Arguing conditionally in a recursive sequence, we can show in the same way as for one-, two- and three-level models, that the general formula for $\mathrm{cov}(U,U)$ is

$$
D = \mathrm{cov}(U,U) = \sum_{\ell=1}^{L} \sigma_{\ell}^{2} \sum_{\mathrm{level}(i)=\ell} \zeta_i (\zeta_i)^{\mathbf{T}} \tag{20}
$$

The inner summation is taken over all clusters i at level ℓ. This can also be written as

$$
D = \Sigma + \sum_{i \text{ not a leaf}} \sigma_{\mathrm{level}(i)}^{2} \zeta_i (\zeta_i)^{\mathbf{T}}
$$

where Σ is a diagonal matrix whose diagonal entry for each leaf r is $\sigma_{\mathrm{level}(r)}^{2}$. If all leaves are at level L, then $\Sigma = \sigma_L^2 I$.

We clearly have

$$
B\zeta_i = \mu^{[i]}
$$

since it is the sum of the vectors $\mu^{[r]}$ for all leaves $r \subseteq i$. The vector ζ_i is the only leaf-vector with this property, since B has full rank. From (9) and (20) it follows that

$$\mathrm{var}(Y) = \mathrm{diag}(\mu) + \sum_{\ell=1}^{L} \sigma_\ell^2 \sum_{\mathrm{level}(i)=\ell} \mu^{[i]} (\mu^{[i]})^{\mathbf{T}} \tag{21}$$

Lower-Level BLUP Formulas and Covariances

By a similar inductive argument, of which the first three levels are given above, the covariance of the level-ℓ random effects can be shown to be

$$D\{\ell\} = \mathrm{var}(U^{\{\ell\}}) = \sum_{\nu=1}^{\ell} \sigma_\nu^2 \sum_{\mathrm{level}(i)=\nu} \zeta_i^{\{\ell\}} (\zeta_i^{\{\ell\}})^{\mathbf{T}} \tag{22}$$

where $\zeta_i^{\{\ell\}}$ is defined analogously to ζ_i, but on the cluster tree truncated at level ℓ, so that the leaves are either of level ℓ or are lower-level leaves of the full tree. Since there are fewer leaves in the truncated tree than in the full L-level tree, the vectors $\{\zeta_i^{\{\ell\}}\}$ are of smaller dimension than the full-tree leaf vectors $\{\zeta_i^{\{L\}}\}$.

The general form of the lower-level covariance matrices is given by (22), where again we write $U^{\{\ell\}}$ for the vector of random effects corresponding to ℓ-leaves. Note that in an unbalanced tree, this vector may contain some of the leaf clusters, even if $\ell \neq L$.

Definition: We define $B^{\{\ell\}}$, the level-ℓ B-matrix, to be the B-matrix defined for the leaves of the level-ℓ trimmed tree. That is, each column of $B^{\{\ell\}}$ corresponds to a leaf r

of the ℓ-trimmed tree, and consists of the Z-vector $\mu^{[r]}$. In the case that r is a leaf of the full tree of level less than ℓ, the corresponding column of $B^{\{\ell\}}$ is identical to that of B.

As before, we define

$$\Delta^{\{\ell\}} = A^{-1}B^{\{\ell\}}$$

which has the same pattern of non-zeros as $B^{\{\ell\}}$, but all non-zeros are 1. We note that $\Delta^{\{\ell\}}U^{\{\ell\}}$ is an expansion of the vector $U^{\{\ell\}}$ into a $\mathbf{Z}$-vector, in the sense that

$$(\Delta^{\{\ell\}}U^{\{\ell\}})^{[e,\tau]} = (U^{\{\ell\}})^{i(e)}$$

where $i(e)$ is the ℓ-leaf containing the individual e.

Lemma:

$$\mathrm{cov}(Y, U^{\{\ell\}}) = B^{\{\ell\}}D^{\{\ell\}}$$

In this notation,

$$\mathrm{cov}(Y, U^{\{L\}}) = BD = B^{\{L\}}D^{\{L\}}$$

Proof:

Let $\kappa = [e,\tau] \in Z$, and suppose that $r = r(e)$ is the leaf-cluster containing e. If $\mathrm{level}(r) > \ell$, then let $i = i(e)$ be the ℓ-leaf ancestral to r. If $\mathrm{level}(r) \le \ell$, then let i be r itself. We first note that if $\pi(r)$ is the parent of r, then $\varepsilon(U^r \mid U^{\pi(r)}) = U^{\pi(r)}$, and by induction we can show that

$$\varepsilon(U^r \mid U^i) = U^i$$

We also note that

$$\varepsilon(Y^\kappa \mid U^i) = \varepsilon(\varepsilon(Y^\kappa \mid U^i, U^r) \mid U^i) = \varepsilon(\mu^\kappa U^r \mid U^i) = \mu^\kappa U^i$$

In matrix terms, this says

$$\varepsilon(Y \mid U) = \mathrm{diag}(\mu)\Delta^{\{\ell\}}U^{\{\ell\}} = A\Delta^{\{\ell\}}U^{\{\ell\}} = B^{\{\ell\}}U^{\{\ell\}}$$

Now we consider $\mathrm{cov}(Y, U^{\{\ell\}})$:

$$\begin{aligned}\mathrm{cov}(Y, U^{\{\ell\}}) &= \varepsilon(\mathrm{cov}(Y, U^{\{\ell\}} \mid U^{\{\ell\}})) + \mathrm{cov}(\varepsilon(Y \mid U^{\{\ell\}}), U^{\{\ell\}}) \\ &= 0 + \mathrm{cov}(B^{\{\ell\}}U^{\{\ell\}}, U^{\{\ell\}}) \\ &= B^{\{\ell\}}D^{\{\ell\}}\end{aligned}$$

This completes the proof.

The BLUP estimator $\hat{U}^{\{\ell\}}$ of $U^{\{\ell\}}$ is again characterized by the orthogonality relation: $U^{\{\ell\}} - \hat{U}^{\{\ell\}}$ is orthogonal to any linear transformation of Y. This implies, by the argument given in the section "BLUP Formula", that

$$\hat{U}^{\{\ell\}} = \varepsilon(U^{\{\ell\}}) + \mathrm{cov}(U^{\{\ell\}}, Y)\,\mathrm{var}(Y)^{-1}(Y - \mu)$$

By again substituting the formulas for $\mathrm{var}(Y)^{-1}$ and $\mathrm{cov}(U^{\{\ell\}}, Y)$, we obtain

$$\hat{U}^{\{\ell\}} = \mathbf{1} + D^{\{\ell\}}(B^{\{\ell\}})^{\mathbf{T}}[A^{-1} - A^{-1}B(I + DQ)^{-1}DB^{\mathbf{T}}A^{-1}](Y - \mu)$$

where here D means $D\{L\}$, and similarly B, Q and A have their original meanings of $B\{L\}$, $Q\{L\}$ and $A\{L\}$. It is straightforward to check that for any leaf r of the full tree, with $\mathrm{level}(r) \le \ell$, the value $(\hat{U}^{\{\ell\}})^r$ is the same for all values of $\ell \le L$. We just carry the lower-level leaves along in the vectors $\hat{U}^{\{\ell\}}$ for convenience.

Now letting

$$P^{\{\ell\}} = (B^{\{\ell\}})^{\mathbf{T}}A^{-1}B,$$

after some algebra, we get the following alternative expression for $\hat{U}^{\ell}$:

$$\begin{aligned}\hat{U}^{\{\ell\}} &= \mathbf{1} + D^{\{\ell\}}(B^{\{\ell\}})^{\mathbf{T}} A^{-1}(Y-\mu) - D^{\{\ell\}} P^{\{\ell\}}[\hat{U}^{\{L\}} - \mathbf{1}] \\ &= \mathbf{1} + D^{\{\ell\}} w^{\{\ell\}} - D^{\{\ell\}} P^{\{\ell\}}[\hat{U}^{\{L\}} - \mathbf{1}],\end{aligned}$$

where $w^{\{\ell\}} = (B^{\{\ell\}})^{\mathbf{T}} A^{-1}(Y-\mu)$. Note that the two $\mathbf{1}$'s in this formula have different dimension. This shows that once we have $\hat{U}^{\{L\}}$, we can get any $\hat{U}^{\{\ell\}}$ without much extra work. We can save even more work by noting that $w^{\{\ell\}}$ is simply an aggregation of $w = (B^{\{L\}})^{\mathbf{T}} A^{-1}(Y-\mu)$: that is, the entry of $w^{\{\ell\}}$ corresponding to level-ℓ cluster i is the sum of the entries of w corresponding to leaves that descend from i.

The matrix $\Gamma^{\{\ell\}}$ that expresses the aggregation has rows corresponding to ℓ-leaves and columns corresponding to level-L leaves. Each row of $\Gamma^{\{\ell\}}$ corresponding to a level-ℓ cluster i has a 1 in each column corresponding to a level-L leaf descending from cluster i; and each row of $\Gamma^{\{\ell\}}$ corresponding to a leaf r of level $< \ell$ has a 1 in the column corresponding to r. To put it another way, each row i is just $(\zeta_i)^{\mathbf{T}}$. We have the properties

$$\begin{aligned}w^{\{\ell\}} &= \Gamma^{\{\ell\}} w \\ B^{\{\ell\}} &= B(\Gamma^{\{\ell\}})^{\mathbf{T}} \\ P^{\{\ell\}} &= \Gamma^{\{\ell\}} Q \\ Q^{\{\ell\}} &= (B^{\{\ell\}})^{\mathbf{T}} A^{-1} B^{\{\ell\}} = \Gamma^{\{\ell\}} Q (\Gamma^{\{\ell\}})^{\mathbf{T}}\end{aligned}$$

and

$$\begin{aligned}\hat{U}^{\{\ell\}} &= \mathbf{1} + D^{\{\ell\}}\Gamma^{\{\ell\}} w - D^{\{\ell\}} P^{\{\ell\}} (I + DQ)^{-1} Dw \\ &= \mathbf{1} + D^{\{\ell\}}[\Gamma^{\{\ell\}} - P^{\{\ell\}} (I + DQ)^{-1} D] w \\ &= \mathbf{1} + D^{\{\ell\}}\Gamma^{\{\ell\}}[I - Q(I + DQ)^{-1} D] w \\ &= \mathbf{1} + D^{\{\ell\}}\Gamma^{\{\ell\}}[I - (Q^{-1} + D)^{-1} D] w \\ \hat{U}^{\{\ell\}} &= \mathbf{1} + D^{\{\ell\}}\Gamma^{\{\ell\}}(Q^{-1} + D)^{-1} Q^{-1} w\end{aligned}$$

Note that everything past the factor $\Gamma^{\{\ell\}}$ is the same as for the leaf-level $\hat{U}$. This means that the lower-level random effects can be obtained for little extra work, once the leaf-level computations are done.

Finally, we use the same short notation as before:

$$\hat{U}^{\{\ell\}} = \mathbf{1} + H^{\{\ell\}}(Y - \mu)$$

where

$$H^{\{\ell\}} = D^{\{\ell\}}(B^{\{\ell\}})^{\mathbf{T}}[A^{-1} - A^{-1}B(I + DQ)^{-1}DB^{\mathbf{T}}A^{-1}]$$

It follows that

$$\text{var}(\hat{U}^{\{\ell\}}) = H^{\{\ell\}}\,\text{var}(Y)(H^{\{\ell\}})^{\mathbf{T}}$$

After some algebra, this becomes

$$\begin{aligned}\text{var}(\hat{U}^{\{\ell\}}) &= D^{\{\ell\}}\Gamma^{\{\ell\}}Q(I + DQ)^{-1}(\Gamma^{\{\ell\}})^{\mathbf{T}}D^{\{\ell\}} \\ &= D^{\{\ell\}}\Gamma^{\{\ell\}}(Q^{-1} + D)^{-1}(\Gamma^{\{\ell\}})^{\mathbf{T}}D^{\{\ell\}}\end{aligned}$$

As already pointed out, $\hat{U}^{\{\ell\}} - U^{\{\ell\}}$ is orthogonal to any linear transformation of Y. We also have

$$\begin{aligned}\text{var}(\hat{U}^{\{\ell\}} - U^{\{\ell\}}) &= -\text{cov}(\hat{U}^{\{\ell\}} - U^{\{\ell\}}, U^{\{\ell\}}) = \text{var}(U^{\{\ell\}}) - \text{var}(\hat{U}^{\{\ell\}}) \\ &= D^{\{\ell\}} - D^{\{\ell\}}\Gamma^{\{\ell\}}(Q^{-1} + D)^{-1}(\Gamma^{\{\ell\}})^{\mathbf{T}}D^{\{\ell\}} \\ &= V^{\{\ell\}}, \text{ say.}\end{aligned} \tag{23}$$

Estimating the σ_ℓ^2

It is easy to produce schemes for estimating the dispersion parameters, but we need to ensure that the estimates are positive. We start with the following observation: let i be a cluster of level $\ell = \ell(i)$ and let $\pi = \pi(i)$ be its parent. Then

$$\begin{aligned}\operatorname{var}(U^i - U^\pi) &= \varepsilon(\operatorname{var}(U^i - U^\pi | U^\pi)) + \operatorname{var}(\varepsilon(U^i - U^\pi | U^\pi)) \\ &= \varepsilon(\operatorname{var}(U^i | U^\pi)) + 0 \\ &= \varepsilon(\sigma_\ell^2 U^\pi) = \sigma_\ell^2\end{aligned}$$

or

$$\operatorname{var}(U^i - U^{\pi(i)}) = \sigma^2_{\text{level}(i)}$$

Letting $d = U^i - U^{\pi(i)}$ and $\hat{d} = \hat{U}^i - \hat{U}^{\pi(i)}$, we have: $\operatorname{cov}(d - \hat{d}, GY) = 0$, G any linear transformation of Y. So

$$\operatorname{cov}(d - \hat{d}, \hat{U}^i) = 0 = \operatorname{cov}(d - \hat{d}, \hat{U}^\pi)$$

and

$$\operatorname{var}(d - \hat{d}) = \operatorname{cov}(d - \hat{d}, d) = \operatorname{var}(d) - \operatorname{cov}(\hat{d}, d) = \operatorname{var}(d) - \operatorname{var}(\hat{d})$$

Also

$$\begin{aligned}\operatorname{var}(d - \hat{d}) &= \operatorname{var}((U^i - \hat{U}^i) - (U^\pi - \hat{U}^\pi)) \\ &= \operatorname{var}(U^i - \hat{U}^i) - 2\operatorname{cov}(U^i - \hat{U}^i, U^\pi - \hat{U}^\pi) + \operatorname{var}(U^\pi - \hat{U}^\pi) \\ &= \operatorname{var}(U^i - \hat{U}^i) - 2\operatorname{cov}(U^i - \hat{U}^i, U^\pi) + \operatorname{var}(U^\pi - \hat{U}^\pi)\end{aligned}$$

$$\begin{aligned}\operatorname{cov}(U^i, U^\pi) &= \varepsilon(\operatorname{cov}(U^i, U^\pi | U^\pi)) + \operatorname{cov}(\varepsilon(U^i | U^\pi), \varepsilon(U^\pi | U^\pi) \\ &= 0 + \operatorname{cov}(U^\pi, U^\pi) = \operatorname{var}(U^\pi)\end{aligned}$$

We also have

$$\begin{aligned}\operatorname{cov}(\hat{U}^{\{\ell\}}, U^{\{\ell-1\}}) &= \operatorname{cov}(\mathbf{1} + D^{\{\ell\}}[\Gamma^{\{\ell\}} - D^{\{\ell\}} P^{\{\ell\}} (I + DQ)^{-1} D]w, U^{\{\ell-1\}}) \\ &= D^{\{\ell\}}[\Gamma^{\{\ell\}} - D^{\{\ell\}} P^{\{\ell\}} (I + DQ)^{-1} D]\operatorname{cov}(w, U^{\{\ell-1\}}) \\ &= D^{\{\ell\}}[\Gamma^{\{\ell\}} - D^{\{\ell\}} P^{\{\ell\}} (I + DQ)^{-1} D]B^{\mathbf{T}} A^{-1} \operatorname{cov}(Y, U^{\{\ell-1\}}) \\ &= D^{\{\ell\}}[\Gamma^{\{\ell\}} - D^{\{\ell\}} P^{\{\ell\}} (I + DQ)^{-1} D]B^{\mathbf{T}} A^{-1} B^{\{\ell-1\}} D^{\{\ell-1\}}\end{aligned}$$

or

$$\begin{aligned}
\operatorname{cov}(\hat{U}^{\{\ell\}}, U^{\{\ell-1\}}) &= D^{\{\ell\}}[\Gamma^{\{\ell\}} - D^{\{\ell\}} P^{\{\ell\}} (I + DQ)^{-1} D] B^{\mathbf{T}} A^{-1} B^{\{\ell-1\}} D^{\{\ell-1\}} \\
&= D^{\{\ell\}} \Gamma^{\{\ell\}} [I - Q(I + DQ)^{-1} D] B^{\mathbf{T}} A^{-1} B (\Gamma^{\{\ell-1\}})^{\mathbf{T}} D^{\{\ell-1\}} \\
&= D^{\{\ell\}} \Gamma^{\{\ell\}} [I - Q(I + DQ)^{-1} D] Q (\Gamma^{\{\ell-1\}})^{\mathbf{T}} D^{\{\ell-1\}} \\
&= D^{\{\ell\}} \Gamma^{\{\ell\}} [Q - Q(I + DQ)^{-1} DQ] (\Gamma^{\{\ell-1\}})^{\mathbf{T}} D^{\{\ell-1\}} \\
&= D^{\{\ell\}} \Gamma^{\{\ell\}} Q (I + DQ)^{-1} (\Gamma^{\{\ell-1\}})^{\mathbf{T}} D^{\{\ell-1\}} \\
&= D^{\{\ell\}} \Gamma^{\{\ell\}} (Q^{-1} + D)^{-1} (\Gamma^{\{\ell-1\}})^{\mathbf{T}} D^{\{\ell-1\}} \\
&= \Psi^{\{\ell\}}, \text{ say.}
\end{aligned}$$

It follows that

$$\begin{aligned}
\operatorname{cov}(U^i - \hat{U}^i, U^\pi) &= \operatorname{cov}(U^i, U^\pi) - \operatorname{cov}(\hat{U}^i, U^\pi) \\
&= \operatorname{var}(U^\pi) - \{\Psi^{\{\ell\}}\}^i_{\pi(i)}
\end{aligned}$$

so finally, using (23)

$$\begin{aligned}
0 < \operatorname{var}(d - \hat{d}) &= \operatorname{var}(U^i - \hat{U}^i) - 2\operatorname{cov}(U^i - \hat{U}^i, U^\pi) + \operatorname{var}(U^\pi - \hat{U}^\pi) \\
&= \{V^{\{\ell\}}\}^i_i - 2[\operatorname{var}(U^\pi) - \{\Psi^{\{\ell\}}\}^i_{\pi(i)}] + \{V^{\{\ell-1\}}\}^\pi_\pi
\end{aligned}$$

where

$$\begin{aligned}
\Psi^{\{\ell\}} &= D^{\{\ell\}} \Gamma^{\{\ell\}} (Q^{-1} + D)^{-1} (\Gamma^{\{\ell-1\}})^{\mathbf{T}} D^{\{\ell-1\}} \\
V^{\{\ell\}} &= D^{\{\ell\}} - D^{\{\ell\}} \Gamma^{\{\ell\}} (Q^{-1} + D)^{-1} (\Gamma^{\{\ell\}})^{\mathbf{T}} D^{\{\ell\}}
\end{aligned}$$

So

$$\begin{aligned}
\operatorname{var}(d) = \sigma^2_\ell &= \operatorname{var}(\hat{d}) + \operatorname{var}(d - \hat{d}) \\
&= \operatorname{var}(\hat{d}) + \{V^{\{\ell\}}\}^i_i - 2[\{D^{\{\ell-1\}}\}^\pi_\pi - \{\Psi^{\{\ell\}}\}^i_\pi] + \{V^{\{\ell-1\}}\}^\pi_\pi
\end{aligned}$$

So the second part is positive, and it follows that this estimating equation for σ^2_ℓ gives positive estimates. Averaging over all the clusters i of level ℓ ($N^{\{\ell\}}$ of them, say), and estimating $\operatorname{var}(\hat{d})$ by the sample variance, we estimate $\hat{\sigma}^2_\ell$ by solving

$$\hat{\sigma}_\ell^2 = (1/N^{\{\ell\}}) \sum_{i=1}^{N^{\{\ell\}}} \Big[(\hat{U}^i - \hat{U}^{\pi(i)})^2 + \{V^{\{\ell\}}\}_i^i - 2\Big(\{D^{\{\ell-1\}}\}_{\pi(i)}^{\pi(i)} - \{\Psi^{\{\ell\}}\}_{\pi(i)}^i\Big) + \{V^{\{\ell-1\}}\}_{\pi(i)}^{\pi(i)} \Big]$$

The right side is also a function of $\hat{\sigma}_\ell^2$ and $\hat{\sigma}_{\ell-1}^2$; so this is an equation which represents the vector of $\{\hat{\sigma}_\ell^2\}$ as a function of the same vector. We will normally use it for Picard iteration, i.e. with new values on the left, and old ones on the right.

Two-Level Distance Decay

Assume there are two levels of clusters: "clusters" and "subclusters" (e.g. SMA's and zip-codes). Clusters are indexed by

$$i = 1, 2, \ldots, m$$

and the subclusters of cluster i are indexed by

$$j = 1, 2, \ldots, J_i$$

Some clusters may have no subclusters, i.e. $J_i = 0$. Let $\mathbf{U}$ denote the vector of cluster-level random effects $\{U_i : i = 1, 2, \ldots, m\}$, and let U denote the vector of subcluster-level random effects, $\{u_{ij} : j = 1, 2, \ldots, J_i, i = 1, 2, \ldots, m\}$. This has dimension $J = \sum J_i$.

The assumptions we make are:

1. $\varepsilon(u_{ij} | U_i) = U_i$

2. $\text{cov}(u_{ij}, u_{ik} | U_i) = 0$ unless subclusters j and k are neighbors (by some definition)

3. $\text{var}(u_{ij}|U_i) = \sigma_{2i}^2 U_i$, where σ_{2i}^2 is a parameter. We will make more explicit covariance assumptions below.

4. (u_{ij}, u_{pq}) are conditionally independent, given (U_i, U_p), if $i \neq p$

5. The matrix $D^{\{1\}} = \text{cov}(U_i, U_p)$ is dense, in general

Let W denote the expected conditional covariance matrix

$$\begin{aligned} W &= \varepsilon_{\mathbf{U}}(\text{cov}(\mathbf{u}, \mathbf{u}|\mathbf{U})) \\ W_{ij,\,pq} &= \varepsilon_{\mathbf{U}}(\varepsilon((u_{ij} - U_i)(u_{pq} - U_p)|U_i, U_p)) \\ &= \begin{cases} 0 & \text{if } i \neq p \\ \varepsilon_{U_i}(\varepsilon((u_{ij} - U_i)(u_{iq} - U_i)|U_i)) & \text{if } i = p \end{cases} \end{aligned}$$

Then the assumptions above imply that W is block-diagonal, with blocks corresponding to the clusters, and block i is $J_i \times J_i$. Now

$$\begin{aligned} \text{cov}(u_{ij}, u_{pq}) &= \varepsilon(\text{cov}(u_{ij}, u_{pq}|U_i, U_p)) + \text{cov}(\varepsilon(u_{ij}|U_i, U_p), \varepsilon(u_{pq}|U_i, U_p)) \\ &= W_{ij,\,pq} + \text{cov}(U_i, U_p) \end{aligned}$$

From this it follows that the full covariance matrix of $\mathbf{u}$ is given by

$$D^{\{2\}} = \text{cov}(\mathbf{u}, \mathbf{u}) = W + \Gamma^{\mathbf{T}} D^{\{1\}} \Gamma = W + \Theta, \tag{24}$$

say, where Γ is a $m \times J$ block-diagonal matrix, the i^{th} block of which is a row of 1's of length J_i.

The numbers mentioned so far are $m = 156$ and J about 3000, so each SMA (cluster) has, on average, about 20 zip codes (i.e. J_i averages about 20). The matrix W can be inverted directly by a Cholesky factorization of each block, and also we can Cholesky-factor $D^{\{1\}}$ as $LL^{\mathbf{T}}$, so we have

$$D^{\{2\}} = W + \Gamma^{\mathbf{T}} L (\Gamma^{\mathbf{T}} L)^{\mathbf{T}}$$

which is in a form suitable for Sherman-Morrison-Woodbury:

$$(A + UV^{\mathrm{T}})^{-1} = A^{-1} - A^{-1}U(I + V^{\mathrm{T}}A^{-1}U)^{-1}V^{\mathrm{T}}A^{-1}$$
$$\text{if } V = U,$$
$$(A + UU^{\mathrm{T}})^{-1} = A^{-1} - A^{-1}U(I + U^{\mathrm{T}}A^{-1}U)^{-1}U^{\mathrm{T}}A^{-1}$$

So, letting $D = D^{\{2\}}$,

$$(Q^{-1} + D)^{-1} = G - G\Gamma^{\mathbf{T}}L\{I + (\Gamma^{\mathbf{T}}L)^{\mathbf{T}}G\Gamma^{\mathbf{T}}L\}^{-1}(\Gamma^{\mathbf{T}}L)^{\mathbf{T}}G$$
$$\text{where}$$
$$G = (Q^{-1} + W)^{-1}$$

However, we may find it simpler to use conjugate gradient directly on the original form (24). In that case, we can relax some of the assumptions above:

1. We don't need much (perhaps not any) sparseness in the blocks of W, so unless an SMA is huge there is no need to restrict the interactions of zip codes to neighbors: nearly any covariance pattern at the subcluster level will do.
2. We can partially relax the assumption (4) of conditional independence, to allow a few direct interactions between zip codes in different SMA's, so long as they are sparse.

This model, for $\mathrm{cov}(\mathbf{u}, \mathbf{u})$ is very feasible for computations, with m = 156 and J = 3000. In fact, we could probably triple these numbers without much trouble, although the volume of linear-algebra calculations slows the program considerably.

Distance-Decay

If the clusters and subclusters each have a distance-decay covariance form, then

$$D^{\{1\}}_{i,p} = \sigma_1^2 \, \rho_1^{d^1_{ip}/h}$$

and

$$W_{ij,\,pq} = \begin{cases} 0 & \text{if } i \neq p \\ \sigma_{2i}^2\, \rho_{2i}^{d_{jq}^2/h} & \text{if } i = p \end{cases}$$

We may impose a condition that all σ_{2i}^2 and ρ_{2i} be equal, but we don't do so here. The distance functions d^1 and d^2 need not be the same, i.e. the distance functions at different levels are completely independent.

Nested Independence

Here we assume that the clusters at each level are conditionally independent, given the U -values at the parent level. It follows that

$$D^{\{1\}} = \sigma_1^2 I$$

and W is diagonal, with $\sigma_{2i}^2 I_i$ on the i^{th} block. We have again

$$\begin{aligned} D^{\{2\}} &= W + \Gamma^{\mathbf{T}} L (\Gamma^{\mathbf{T}} L)^{\mathbf{T}} \\ &= W + \sigma_1^2 \Gamma^{\mathbf{T}} \Gamma \end{aligned}$$

and $\Gamma^{\mathbf{T}}\Gamma$ is block-diagonal with each block consisting of all 1's.

Now letting $D = D^{\{2\}}$ and $H = (Q^{-1} + D)^{-1}$,

$$H = (Q^{-1} + D)^{-1} = G - \sigma_1^2 G \Gamma^{\mathbf{T}} \{ I + \sigma_1^2 \Gamma G \Gamma^{\mathbf{T}} \}^{-1} \Gamma G$$

where

$$G = (Q^{-1} + W)^{-1}$$

which is very easy to apply because $Q^{-1} + W$ and $\Gamma (Q^{-1} + W)^{-1} \Gamma^{\mathbf{T}}$ are diagonal.

Clusters Without Subclusters

If there is a cluster p not subdivided into subclusters, then all subcluster values $\{u_{ij}\}$ are conditionally independent of U_p, given all $\{U_i \,|\, i \neq p\}$. This means that the matrix W

has a diagonal block corresponding to p, which is 1×1. Also p is a leaf cluster, so should appear in the list of leaves. The simplest way to do this is to give p a fictional subcluster $p1$, consisting of the whole cluster p. From the definition of W it follows that the diagonal entry $W_{p1,p1} = 0$. From the formulas for $\hat{\mathbf{u}}$ and $\hat{\mathbf{U}}$ given below in (26) and (25), a simple algebraic argument shows that $\hat{u}_{p1} = \hat{U}_p$, and so for convenience we can carry u_{p1} along in the computations without fear of inconsistency.

Parameter Estimation

General Approach

Let $D = D^{\{2\}}$ be the covariance matrix at level 2. Letting

$$\begin{aligned} Q &= B^{\mathrm{T}} A^{-1} B \\ w &= B^{\mathrm{T}} A^{-1} (Y - \mu) \end{aligned}$$

Q is a diagonal matrix depending on $\mu = \exp(\mathbf{X}\gamma)$.

$$\begin{aligned} \operatorname{var}(w) &= Q + Q\,D\,Q \\ Q^{-1} \operatorname{var}(w) Q^{-1} &= Q^{-1} + D \end{aligned}$$

The BLUP formulas give

$$\begin{aligned} \hat{\mathbf{u}} &= \mathbf{1} + Q^{-1}(Q^{-1} + D)^{-1} D w \\ &= \mathbf{1} + D(Q^{-1} + D)^{-1} Q^{-1} w \\ \operatorname{var}(\hat{\mathbf{u}}) &= D(Q^{-1} + D)^{-1} Q^{-1} \operatorname{var}(w) Q^{-1} (Q^{-1} + D)^{-1} D \\ &= D(Q^{-1} + D)^{-1} D \\ D &= \operatorname{var}(\mathbf{u}) = \operatorname{var}(\hat{\mathbf{u}}) + \operatorname{var}(\mathbf{u} - \hat{\mathbf{u}}) \\ &= D(Q^{-1} + D)^{-1} D + \operatorname{var}(\mathbf{u} - \hat{\mathbf{u}}) \end{aligned} \tag{25}$$

$$\begin{aligned}\text{var}(\mathbf{u}-\hat{\mathbf{u}}) &= \text{var}(\mathbf{u}) - \text{var}(\hat{\mathbf{u}}) = D - D(D+Q^{-1})^{-1}D \\ &= Q^{-1}(D+Q^{-1})^{-1}D\end{aligned}$$

Also:

$$\begin{aligned}\hat{\mathbf{u}} &= \mathbf{1} + D(Q^{-1}+D)^{-1}Q^{-1}w \\ \hat{\mathbf{U}} &= \mathbf{1} + D^{\{1\}}\Gamma(Q^{-1}+D)^{-1}Q^{-1}w\end{aligned} \tag{26}$$

and

$$\begin{aligned}\text{var}(\hat{\mathbf{U}}) &= D^{\{1\}}\Gamma(Q^{-1}+D)^{-1}Q^{-1}[Q+QDQ]Q^{-1}(Q^{-1}+D)^{-1}\Gamma^{\mathbf{T}}D^{\{1\}} \\ &= D^{\{1\}}\Gamma(Q^{-1}+D)^{-1}[Q^{-1}+D](Q^{-1}+D)^{-1}\Gamma^{\mathbf{T}}D^{\{1\}} \\ &= D^{\{1\}}\Gamma(Q^{-1}+D)^{-1}\Gamma^{\mathbf{T}}D^{\{1\}} \\ \text{var}(\mathbf{U}-\hat{\mathbf{U}}) &= \text{var}(\mathbf{U}) - \text{var}(\hat{\mathbf{U}}) = D^{\{1\}} - D^{\{1\}}\Gamma(Q^{-1}+D)^{-1}\Gamma^{\mathbf{T}}D^{\{1\}} \\ &= D^{\{1\}}[I - \Gamma(Q^{-1}+D)^{-1}\Gamma^{\mathbf{T}}D^{\{1\}}] \\ \text{var}(\Gamma^{\mathbf{T}}\mathbf{U} - \Gamma^{\mathbf{T}}\hat{\mathbf{U}}) &= \Gamma^{\mathbf{T}}D^{\{1\}}\Gamma - \Gamma^{\mathbf{T}}D^{\{1\}}\Gamma(Q^{-1}+D)^{-1}\Gamma^{\mathbf{T}}D^{\{1\}}\Gamma = V, \text{ say}\end{aligned}$$

We have

$$\text{var}(u_{ij}|U_i) = \sigma_{2i}^2 U_i$$

and

$$\begin{aligned}\text{var}(u_{ij}-U_i) &= \varepsilon(\text{var}(u_{ij}-U_i|U_i)) + \text{var}(\varepsilon(u_{ij}-U_i|U_i)) \\ &= \varepsilon(\text{var}(u_{ij}|U_i)) + 0 \\ &= \sigma_{2i}^2 = W_{ij,\,ij} \\ \text{cov}(u_{ij}-U_i, u_{pk}-U_p) &= \varepsilon(\text{cov}(u_{ij}-U_i, u_{pk}-U_p|U_i,U_p)) + \text{cov}(\varepsilon(u_{ij}-U_i|U_i), \varepsilon(u_{pk}-U_p|U_p)) \\ &= \begin{cases} 0 & \text{if} \quad i \neq p \\ W_{ij,\,ik} & \text{if} \quad i = p \end{cases}\end{aligned}$$

so finally

$$\text{var}(\mathbf{d}) = \text{var}(\mathbf{u} - \Gamma^{\mathbf{T}}\mathbf{U}) = W$$

where we let $d_{ij} = u_{ij} - U_i$, or $\mathbf{d} = \mathbf{u} - \Gamma^{\mathbf{T}}\mathbf{U}$. We note that

$$W = \text{var}(\mathbf{d}) = \text{var}(\hat{\mathbf{d}}) + \text{var}(\mathbf{d} - \hat{\mathbf{d}})$$

and we need a formula for $\text{var}(\mathbf{d} - \hat{\mathbf{d}})$:

$$\text{var}(d_{ij} - \hat{d}_{ij}) = \text{var}((u_{ij} - U_i) - (\hat{u}_{ij} - \hat{U}_i)) = \text{var}((u_{ij} - \hat{u}_{ij}) - (U_i - \hat{U}_i))$$
$$\text{var}(d_{ij} - \hat{d}_{ij}) = \text{var}(d_{ij}) - \text{var}(\hat{d}_{ij}) = \text{var}(u_{ij} - \hat{u}_{ij}) - 2\,\text{cov}(u_{ij} - \hat{u}_{ij}, U_i) + \text{var}(U_i - \hat{U}_i)$$
$$\text{var}(d_{ij} - \hat{d}_{ij}) = \text{var}(u_{ij} - \hat{u}_{ij}) - 2\,\text{cov}(u_{ij} - \hat{u}_{ij}, U_i) + \text{var}(U_i - \hat{U}_i)$$

So now we need a formula for $\text{cov}(u_{ij} - \hat{u}_{ij}, U_i)$. We have

$$\text{cov}(u_{ij}, U_i) = \varepsilon(\text{cov}(u_{ij}, U_i | U_i)) + \text{cov}(\varepsilon(u_{ij} | U_i), \varepsilon(U_i | U_i)) = 0 + \text{var}(U_i)$$
$$\text{cov}(u_{ij}, U_p) = \text{cov}(U_i, U_p), \;\; p \neq i$$

or

$$\text{cov}(\mathbf{u}, \mathbf{U}) = \Gamma^{\mathbf{T}} \text{var}(\mathbf{U}) = \Gamma^{\mathbf{T}} D^{\{1\}}$$
$$\text{cov}(\mathbf{u}, \Gamma^{\mathbf{T}}\mathbf{U}) = \Gamma^{\mathbf{T}} D^{\{1\}} \Gamma$$

and

$$\begin{aligned}
\text{cov}(\hat{\mathbf{u}}, \mathbf{U}) &= \text{cov}(\mathbf{1} + Q^{-1}(Q^{-1} + D)^{-1} Dw, \mathbf{U}) \\
&= Q^{-1}(Q^{-1} + D)^{-1} D\,\text{cov}(w, \mathbf{U}) \\
&= Q^{-1}(Q^{-1} + D)^{-1} DB^{\mathbf{T}} A^{-1}\,\text{cov}(Y, \mathbf{U}) \\
&= Q^{-1}(Q^{-1} + D)^{-1} DB^{\mathbf{T}} A^{-1} B^{\{1\}} D^{\{1\}} \\
&= Q^{-1}(Q^{-1} + D)^{-1} DB^{\mathbf{T}} A^{-1} B \Gamma^{\mathbf{T}} D^{\{1\}} \\
&= Q^{-1}(Q^{-1} + D)^{-1} DQ\Gamma^{\mathbf{T}} D^{\{1\}}
\end{aligned}$$

So, we obtain

$$\begin{aligned}
\text{cov}(\mathbf{u} - \hat{\mathbf{u}}, \mathbf{U}) &= \Gamma^{\mathbf{T}} D^{\{1\}} - Q^{-1}(Q^{-1} + D)^{-1} DQ\Gamma^{\mathbf{T}} D^{\{1\}} \\
&= (I - Q^{-1}(Q^{-1} + D)^{-1} DQ)\Gamma^{\mathbf{T}} D^{\{1\}} \\
&= (I + DQ)^{-1} \Gamma^{\mathbf{T}} D^{\{1\}} \\
&= Q^{-1}(Q^{-1} + D)^{-1} \Gamma^{\mathbf{T}} D^{\{1\}} \\
\text{cov}(\mathbf{u} - \hat{\mathbf{u}}, \Gamma^{\mathbf{T}}\mathbf{U}) &= Q^{-1}(Q^{-1} + D)^{-1} \Gamma^{\mathbf{T}} D^{\{1\}} \Gamma \\
&= Q^{-1} H \Gamma^{\mathbf{T}} D^{\{1\}} \Gamma \\
&= Q^{-1} H \Theta
\end{aligned}$$

where

$$H = (Q^{-1} + D)^{-1}$$

Let

$$\Theta = \Gamma^{\mathbf{T}} D^{\{1\}} \Gamma = \Gamma^{\mathbf{T}} L (\Gamma^{\mathbf{T}} L)^{\mathbf{T}} = \Gamma^{\mathbf{T}} L L^{\mathbf{T}} \Gamma$$

note that Θ is constant on each block (i, p). Let

$$\begin{aligned} G &= (Q^{-1} + W)^{-1} \\ H &= (Q^{-1} + D)^{-1} = (Q^{-1} + W + \Theta)^{-1} \\ &= (Q^{-1} + W + \Gamma^{\mathbf{T}} L (\Gamma^{\mathbf{T}} L)^{\mathbf{T}})^{-1} \\ H &= G - G\Gamma^{\mathbf{T}} L [I + L^{\mathbf{T}} \Gamma G \Gamma^{\mathbf{T}} L]^{-1} L^{\mathbf{T}} \Gamma G \end{aligned}$$

Also

$$\begin{aligned} H &= (Q^{-1} + D)^{-1} = (Q^{-1} + W + \Theta)^{-1} \\ &= (Q^{-1} + W + \Gamma^{\mathbf{T}} D^1 \Gamma)^{-1} \\ (A + UV^{\mathbf{T}})^{-1} &= A^{-1} - A^{-1} U (I + V^{\mathbf{T}} A^{-1} U)^{-1} V^{\mathbf{T}} A^{-1} \end{aligned}$$

so

$$\left\{ \begin{matrix} A = Q^{-1} + W \\ U = \Gamma^{\mathbf{T}} D^1 \\ V = \Gamma^{\mathbf{T}} \end{matrix} \right\} \Rightarrow$$

$$\begin{aligned} H &= G - G\Gamma^{\mathbf{T}} D^1 (I + \Gamma G \Gamma^{\mathbf{T}} D^1)^{-1} \Gamma G \\ &= G - G\Gamma^{\mathbf{T}} D^1 (I + C D^1)^{-1} \Gamma G \end{aligned}$$

Now: we have

$$\Gamma G \Gamma^{\mathbf{T}} = C = \text{diag}(\{\gamma_p\})$$

where $\gamma_p = \mathbf{1}_p^{\mathbf{T}} G_p \mathbf{1}_p$, the sum of all entries of G_p. This gives

$$\begin{aligned} H &= G - G\Gamma^{\mathbf{T}} L [I + L^{\mathbf{T}} C L]^{-1} L^{\mathbf{T}} \Gamma G \\ H &= G - G\Gamma^{\mathbf{T}} Z \Gamma G \\ &\text{where} \\ Z &= L [I + L^{\mathbf{T}} C L]^{-1} L^{\mathbf{T}} \\ &= [L^{-\mathbf{T}} L^{-1} + C]^{-1} \\ &= [(D^{\{1\}})^{-1} + C]^{-1} \\ &= (I + C D^{\{1\}})^{-1} D^{\{1\}} \\ Z &= C^{-1} (C^{-1} + D^{\{1\}})^{-1} D^{\{1\}} \end{aligned}$$

and ΓG can be constructed from the vector $G\mathbf{1}$.

Then

$$\text{var}(d_{ij} - \hat{d}_{ij}) = \text{var}(u_{ij} - \hat{u}_{ij}) - 2\,\text{cov}(u_{ij} - \hat{u}_{ij}, U_i) + \text{var}(U_i - \hat{U}_i)$$

or

$$\text{var}(\mathbf{d} - \hat{\mathbf{d}}) = \text{var}((\mathbf{u} - \hat{\mathbf{u}}) - \Gamma^{\mathbf{T}}(\mathbf{U} - \hat{\mathbf{U}}))$$

$$\text{var}(\mathbf{d} - \hat{\mathbf{d}}) = \text{var}(\mathbf{u} - \hat{\mathbf{u}}) - \text{cov}(\mathbf{u} - \hat{\mathbf{u}}, \Gamma^{\mathbf{T}}\mathbf{U}) - \text{cov}(\Gamma^{\mathbf{T}}\mathbf{U}, \mathbf{u} - \hat{\mathbf{u}}) + \text{var}(\Gamma^{\mathbf{T}}\mathbf{U} - \Gamma^{\mathbf{T}}\hat{\mathbf{U}})$$

since

$$\text{cov}(\mathbf{u} - \hat{\mathbf{u}}, \mathbf{A}\hat{\mathbf{U}}) = 0$$

for any matrix $\mathbf{A}$; so

$$\begin{aligned}
\text{var}(\mathbf{d} - \hat{\mathbf{d}}) &= \text{var}(\mathbf{u} - \hat{\mathbf{u}}) - Q^{-1}H\Theta - \Theta HQ^{-1} + \text{var}(\Gamma^{\mathbf{T}}\mathbf{U} - \Gamma^{\mathbf{T}}\hat{\mathbf{U}}) \\
&= Q^{-1}HD - Q^{-1}H\Theta - \Theta HQ^{-1} + \Gamma^{\mathbf{T}}\,\text{var}(\mathbf{U} - \hat{\mathbf{U}})\Gamma \\
&= Q^{-1}HD - Q^{-1}H\Theta - \Theta HQ^{-1} + \Theta - \Theta H\Theta \\
&= P - \Psi - \Psi^{\mathbf{T}} + V
\end{aligned}$$

Then,

$$\begin{aligned}
P &= DHQ^{-1} = Q^{-1}HD = Q^{-1}H(W + \Theta) = \\
\Psi &= Q^{-1}H\Gamma^{\mathbf{T}}D^{\{1\}}\Gamma = Q^{-1}H\Theta \\
\Psi^{\mathbf{T}} &= \Gamma^{\mathbf{T}}D^{\{1\}}\Gamma HQ^{-1} \\
V &= \Gamma^{\mathbf{T}}D^{\{1\}}\Gamma - \Gamma^{\mathbf{T}}D^{\{1\}}\Gamma H\Gamma^{\mathbf{T}}D^{\{1\}}\Gamma = \Theta - \Theta H\Theta \\
&= \Gamma^{\mathbf{T}}[D^{\{1\}} - D^{\{1\}}\Gamma H\Gamma^{\mathbf{T}}D^{\{1\}}]\Gamma \\
&= \Gamma^{\mathbf{T}}\,\text{var}(\mathbf{U} - \hat{\mathbf{U}})\Gamma
\end{aligned}$$

Now:

$$\begin{aligned}
W &= \text{var}(\mathbf{d}) = \text{var}(\hat{\mathbf{d}}) + \text{var}(\mathbf{d} - \hat{\mathbf{d}}) \\
&= \text{var}(\hat{\mathbf{d}}) + [P - \Psi - \Psi^{\mathbf{T}} + V]
\end{aligned}$$

then

$$\begin{aligned}
\text{var}(\mathbf{d} - \hat{\mathbf{d}}) &= Q^{-1}HD - Q^{-1}H\Theta - \Theta HQ^{-1} + \Theta - \Theta H\Theta \\
&= Q^{-1}H(W + \Theta) - Q^{-1}H\Theta - \Theta HQ^{-1} + \Theta - \Theta H\Theta \\
&= Q^{-1}H(Q^{-1} + W + \Theta) - Q^{-1}HQ^{-1} - Q^{-1}H\Theta - \Theta HQ^{-1} + \Theta - \Theta H\Theta \\
\text{var}(\mathbf{d} - \hat{\mathbf{d}}) &= Q^{-1} - Q^{-1}HQ^{-1} - Q^{-1}H\Theta - \Theta HQ^{-1} + \Theta - \Theta H\Theta \\
&= P - \Psi - \Psi^{\mathbf{T}} + V
\end{aligned}$$

where

$$H = (Q^{-1} + D)^{-1} = (Q^{-1} + W + \Theta)^{-1}$$
$$\Theta = \Gamma^{\mathbf{T}} D^{\{1\}} \Gamma = \Gamma^{\mathbf{T}} L (\Gamma^{\mathbf{T}} L)^{\mathbf{T}} = \Gamma^{\mathbf{T}} L L^{\mathbf{T}} \Gamma$$

Since W is block-diagonal, the off-block-diagonal entries of var($\hat{\mathbf{d}}$) and $[P - \Psi - \Psi^{\mathbf{T}} + V]$ cancel. Now let

$$K = (\hat{\mathbf{u}} - \Gamma \mathbf{U})(\hat{\mathbf{u}} - \Gamma \mathbf{U})^{\mathbf{T}} + [P - \Psi - \Psi^{\mathbf{T}} + V]$$

We consider only the block-diagonal entries of K, i.e. block i is

$$K^i = (\hat{\mathbf{u}}_i - \hat{U}_i)(\hat{\mathbf{u}}_i - \hat{U}_i)^{\mathbf{T}} + [P - \Psi - \Psi^{\mathbf{T}} + V]^i$$

where $\hat{\mathbf{u}}_i$ is the J_i-vector of entries $\{u_{ij}\}$, and $[P - \Psi - \Psi^{\mathbf{T}} + V]^i$ is the $J_i \times J_i$ block corresponding to cluster i. Then we want to choose the parameters so that K and W are similar, say in the Frobenius norm. Now we are postulating that

$$W_{ij,\,pq} = \begin{cases} 0 & \text{if } i \neq p \\ \sigma_{2i}^2 \, \rho_{2i}^{d_{jq}^2 / h} & \text{if } i = p \end{cases}$$

So we let W^i be the block i of W, and considering first the diagonal, minimize

$$\| \operatorname{diag}(K) - \operatorname{diag}(W) \|^2 = \sum_{j=1}^{J_i} (K_{jj}^i - \sigma_{2i}^2)^2$$

with respect to σ_{2i}^2. The minimum is clearly achieved by

$$\hat{\sigma}_{2i}^2 = (1 / J_i) \sum_{j=1}^{J_i} K_{jj}^i$$

Now consider the off-diagonal entries, taking σ_{2i}^2 as given. We minimize

$$e(\rho) = \tfrac{1}{2} \| \operatorname{offdiag}(K) - \operatorname{offdiag}(W) \|^2 = \sum_{j=1}^{J_i} \sum_{q=1}^{j-1} (K_{jq}^i - \sigma_{2i}^2 \, \rho_{2i}^{d_{jq}^2})^2$$

This can be minimized in various ways, and the result is $\hat{\rho}_{2i}$. Note, of course, that K is a function of the dispersion parameters, so these formulas for σ_{2i}^2 and ρ_{2i} define estimating equations that must be solved by iteration.

To estimate the first-level parameters σ_1^2 and ρ_1, we use

$$\begin{aligned}\text{var}(\mathbf{U}-\hat{\mathbf{U}}) &= \text{var}(\mathbf{U}) - \text{var}(\hat{\mathbf{U}}) = D^{\{1\}} - D^{\{1\}}\Gamma H \Gamma^{\mathbf{T}} D^{\{1\}} \\ &= D^{\{1\}}[I - \Gamma H \Gamma^{\mathbf{T}} D^{\{1\}}]\end{aligned}$$

$$\begin{aligned}D^{\{1\}} &= \text{var}(\mathbf{U}) = \text{var}(\hat{\mathbf{U}}) + \text{var}(\mathbf{U}-\hat{\mathbf{U}}) \\ &= \text{var}(\hat{\mathbf{U}}) + D^{\{1\}}[I - \Gamma H \Gamma^{\mathbf{T}} D^{\{1\}}]\end{aligned}$$

So let

$$L = (\hat{\mathbf{U}} - \mathbf{1})(\hat{\mathbf{U}} - \mathbf{1})^{\mathbf{T}} + D^{\{1\}} - D^{\{1\}}\Gamma H \Gamma^{\mathbf{T}} D^{\{1\}}$$

The second term is formed from the current estimates of all dispersion parameters. We now proceed as in the one-level case, to choose the parameters $\hat{\sigma}_1^2$ and $\hat{\rho}_1$ to minimize the Frobenius norm of $L - D^{\{1\}}(\sigma^2, \rho)$.

Appendix: Algorithms

The Problem

On each iteration, we must compute the approximate Schur complement matrix $\hat{K}$, the right side vector $q = L_\beta$, the diagonal matrix Q, and several other quantities. At the end of the iterations, we must compute the exact Schur complement matrix K, to obtain standard errors of the regression coefficients. The formulas of section "Derivatives of log(ℓ)" suggest the difficulty in computing these quantities: they involve a summation over the risk-set index vector *Z*, which can be very large. To illustrate the problem and the algorithms that solve it, we will take as an example the right side vector q; the algorithms for computing the other quantities (the terms of $\hat{K}$) are variants of the ones given here for q.

There are several ways to write the vector q: as we have seen,

$$\begin{aligned} q &= L_\beta \\ &= \widetilde{\mathbf{R}}^{\mathbf{T}}[Y - B_{old}\hat{U}_{old}] \\ &= \sum_{s=1}^{a}\sum_{h=1}^{q}\left[\sum_{k\in R_{sh}} \chi^k \mathbf{R}^k - \exp(\alpha^{sh}) \sum_{k\in R_{sh}} \hat{U}^{r^k} \exp(\mathbf{R}^k\beta)\mathbf{R}^k\right] \end{aligned}$$

This last comes from section "Derivatives of log(ℓ)". The first term of q is easy, since $\chi^k = 0$ unless there is an event at t^k_{end}. So we can simply step through the data matrix $\mathbf{M}$, adding covariate rows $\mathbf{R}^k$ for the records with an event. The total time for this is proportional to the numbers $N_{\mathbf{M}}$ of rows of $\mathbf{M}$, which is optimal. The second term is more difficult, since as pointed out in section "*Z* -Vectors", a summation of the form $\sum_{s=1}^{a}\sum_{h=1}^{q}\sum_{k\in R_{sh}}$ is the same as a summation over *Z*, something we want to avoid, as we

can have $N_Z / N_{\mathbf{M}} \geq 90$. Let us rewrite the expression above as (letting q_2 denote the second term of q)

$$q_2 = -\sum_{s=1}^{a}\sum_{h=1}^{q}\exp(\alpha^{sh})\left[\sum_{k\in R_{sh}}\hat{U}^{r^k}\exp(\mathbf{R}^k\beta)\mathbf{R}^k\right]$$
$$= -\sum_{r=1}^{N_{leaf}}\hat{U}^{r^k}\sum_{s=1}^{a}\sum_{h=1}^{q}\exp(\alpha^{sh})\left[\sum_{k\in R_{sh}\cap r}\exp(\mathbf{R}^k\beta)\mathbf{R}^k\right]$$

We will first focus on sums of the form

$$S_r = \sum_{s=1}^{a}\sum_{h=1}^{q_s}a^{sh}\sum_{k\in R_h\ \&\ r^k=r}b^k$$

where a and b can be scalar, vector, or matrix-valued, so long as the product is conformable. The vector q_2 is a sum over leaf-clusters of terms of this form. We want to be able to compute such expressions S_r for *all* leaf-clusters r, in time proportional to $N_{\mathbf{M}}$, or as close as we can come to this bound.

Now we assumed the rows of $\mathbf{M}$ are ordered by τ_{start} within τ_{end} within stratum s. We assume that, for each record (i.e. row) k of $\mathbf{M}$, the corresponding individual is at risk of an event in the interval $(\tau^k_{start}, \tau^k_{end}]$, and the event occurs at τ^k_{end} if at all in this interval. Whether the event occurs at τ^k_{end} or not is indicated by the value of χ^k. Recall the list $\mathbf{F}_s = \{\tau_{s1}, \tau_{s2}, \tau_{s3}, \ldots, \tau_{sq_s}\}$ of sorted distinct event-times in each stratum. Note the difference: the values $\{\tau^k_{end}\}$ are *all* the τ_{end} times, whether event or censoring; the values $\{\tau_{sh}\}$ are just the event-times.

Definition: For each event-time τ_{sh} define the "delete list" d_{sh} as (for $h = 1, \cdots, q_s - 1$)

$$
\begin{aligned}
d_{sh} &= \{k \in s | k \in R(\tau_{h+1}) \;\&\; k \notin R(\tau_{sh})\} \\
&= \{k \in s | k \in R(\tau_{s(h+1)}) \;\&\; \tau_{sh} \le \tau^k_{start}\} \\
&= \{k \in s |\; \tau_{sh} \le \tau^k_{start} < \tau_{s(h+1)} \le \tau^k_{end}\} \\
\text{if } (\tau^k_{start} == 0,\ \forall k),\ d_{sh} &= \emptyset,\ \text{the empty set}
\end{aligned}
$$

For each s and h, the delete-list d_{sh} consists of those rows k in stratum s that *leave* the risk-set at τ_{sh}, as we step *backward* in time through the event-time list. The sets F_s and d_{sh} can all be formed with a single backward pass through $\mathbf{M}$.

The idea of the algorithms to follow is simple: in each stratum, we step backward through the stratum's rows in $\mathbf{M}$, keeping a running tally of the risk set, adding rows as they enter the risk set, and deleting them as they leave. The sum of b^k over the risk set is updated, rather than fully summed anew for each event-time. This is the source of the saving.

No Secondary Table

We first assume there is no secondary data table, so either there are no time-dependent covariates, or the time-dependence is represented by repeating rows of $\mathbf{M}$.

The following algorithm computes sums of the form

$$
S_r = \sum_{s=1}^{a} \sum_{h=1}^{q_s} a^{sh} \sum_{k \in R_{sh} \;\&\; r^k = r} b^k
$$

for all leaf clusters r, for any values a^{sh} defined on the list $\{\tau_{s1}, \tau_{s2}, \tau_{s3}, \ldots, \tau_{sq_s}\}$ of event times, and for any values b^k computable from the rows of $\mathbf{M}$. Since strata do not really enter into the algorithm, we state it for one stratum and drop the index s. Accomodating stratification is just a matter of summing over s afterward.

Algorithm **1**: To form the sums S_r as defined above, for all leaf clusters r.

Input: the matrix $\mathbf{M}$, the list $\{\tau_1, \tau_2, \tau_3, \ldots, \tau_q\}$ of event times, the lists $\{d_h\}$ of delete-lists, and the values $\{a^h\}$, for all $h = 1, \ldots, q$, and $\{b^k\}$ for all rows k of $\mathbf{M}$ (or a way of computing a^h and b^k on the fly)

Initialize $S_r = 0$, $P^r = 0$, for all leaf clusters r; $k =$ last row of $\mathbf{M}$ in ordering given above.

Step $h = q, \ldots, 1$

While $\tau_h \le \tau^k_{end}$

If ($\tau^k_{start} < \tau_h$) // if k is in the risk set of τ_h

$P^{r^k} = P^{r^k} + b^k$ // Add new values that enter the pool in $(\tau_h, \tau_{h+1}]$

EndIf

$k = k - 1$

EndWhile (on k)

Step ν through d_h

$P^{r^\nu} = P^{r^\nu} - b^\nu$ // Subtract old values that leave the pool in $(\tau_h, \tau_{h+1}]$

EndLoop (on ν)

Step r through the set of leaf clusters

$S_r = S_r + a^h P^r$

EndLoop (on r)

EndLoop (on h)

In this algorithm, the value of k is stepped from $N_{\mathbf{M}}$ down to 1, and the value of ν is stepped through all the delete-lists, which are disjoint. It follows that the time required for Algorithm 1 is proportional to $N_{\mathbf{M}} + N_\tau N_{leaf}$, where N_τ is the total number of distinct event-times per stratum, totalled over all strata. The term $N_\tau N_{leaf}$ is often much smaller than $N_{\mathbf{M}}$, although it can be much larger; it comes from the final loop on r.

There are several variants of Algorithm 1 that are used: for example, sometimes we want to sum on r, but not on h, producing expressions of the form

$$V_h = \sum_{s=1}^{a} a^{sh} \sum_{k \in R_{sh}} b^k$$

It is easy to modify the above algorithm to produce these terms: we just don't split out the clusters r, and don't sum on h; the algorithm requires time proportional to $N_{\mathbf{M}}$. We can also sum on both h and r: the vector q_2 is actually of that form. We showed the algorithm that splits out r for illustration. It is necessary only for computing the diagonal matrix Q.

It is also possible, at the cost of additional complexity, to modify Algorithm 1 so that its run-time is proportional to $N_{\mathbf{M}} + N_\tau + N_{leaf}$. This can be a considerable improvement if the number of strata and the number of leaf clusters are large; for the ACS data, for which $N_{\mathbf{M}}$ is about half a million, values of N_τ can be on the order of 18,000 and N_{leaf} on the order of 10,000 for the most ambitious models. Clearly we prefer the run-time to depend on their sum rather than their product! The modified algorithm is based on the principle that on one pass through the h-loop, not too many leaf-clusters (r) will be encountered, but in Algorithm 1, all leaf-clusters are updated on every h-value. Instead,

we can keep track, for each leaf r, of the h-value on which it was last updated; each time r is encountered, we do a batch update for all h-values since the last update. The algorithm is as follows, again for sums

$$S_r = \sum_{s=1}^{a} \sum_{h=1}^{q_s} a^{sh} \sum_{k \in R_{sh} \,\&\, r^k = r} b^k$$

Algorithm **2**: To form the sums S_r as defined above, for all leaf clusters r. (Modified version, faster for large problems)

Input: the matrix $\mathbf{M}$, the list $\{\tau_1, \tau_2, \tau_3, \ldots, \tau_q\}$ of event times, the lists $\{d_h\}$ of delete-lists, and the values $\{a^h\}$, for all $h = 1, \ldots, q$, and $\{b^k\}$ for all rows k of $\mathbf{M}$ (or a way of computing a^h and b^k on the fly), and the cumulative sums $C_h = \sum_{j=h}^{q} a^j$; we define $C_{q+1} = 0$.

Initialize $S_r = 0$, $P^r = 0$, $V[r] = q+1$, for all leaf clusters r; and $k =$ last row of $\mathbf{M}$ in ordering given above.

Step $h = q, \ldots, 1$

While $\tau_h \le \tau_{end}^k$

If ($\tau_{start}^k < \tau_h$) // if k is in the risk set of τ_h

$d = C_h - C_{V[r^k]}$

$S_{r^k} = S_{r^k} + d P^{r^k} + a^h b^k$

$P^{r^k} = P^{r^k} + b^k$ // Add new values that enter the pool in $(\tau_h, \tau_{h+1}]$

$V[r^k] = h$ // Last h on which r^k is updated

EndIf

$k = k - 1$

EndWhile (on k)

Step ν through d_h

$d = C_h - C_{V[r^\nu]}$

$S_{r^\nu} = S_{r^\nu} + d\, P^{r^\nu}$

$P^{r^\nu} = P^{r^\nu} - b^\nu$ // Subtract old values that leave the pool in $(\tau_h, \tau_{h+1}]$

$V[r^\nu] = h$

EndLoop (on ν)

EndLoop (on h)

Step r through the set of leaf clusters // Cleanup loop

$d = C_1 - C_{V[r]}$

$S_r = S_r + d\, P^r$

EndLoop (on r)

Algorithm 1 or 2, and their variants, are sufficient for computing all the terms of the equation $\hat{K}\,\Delta\beta = q$, for the vector $B^{\mathbf{T}} A^{-1} \mu$ (the diagonal of the matrix Q), and for the vector α, all in time proportional to $N_{\mathbf{M}} + N_\tau\, N_{leaf}$ or $N_{\mathbf{M}} + N_\tau + N_{leaf}$. Naive algorithms would require time proportional to $N_{\mathbf{z}}$.

With Secondary Table

The defining feature of a secondary table is that the conceptual covariate matrix $\widetilde{\mathbf{R}}$ has a set of non-time-dependent columns, and a set of time-dependent ones. Let the key-

variable be denoted by ξ; we assume that each individual has a ξ-value $\xi(e)$ associated. For example, if ξ represents cities, then $\xi(e)$ is the city of residence of the individual e. We can of course have $\xi(e) = e$, i.e. that the table is keyed by individual. Under the assumptions, each row $\widetilde{\mathbf{R}}[e, \tau_h]$ of $\widetilde{\mathbf{R}}$ has two parts: the first part depends on individual, but not on time, and the second part depends on time, but on individual only through the key-variable $\xi(e)$. If, for example, the second part is an air-pollution reading from a single monitor located in each city, then the values depend on individuals only through their city of residence.

$$\widetilde{\mathbf{R}}[e, \tau_h] = \left[\widetilde{\mathbf{R}}_1[e] \quad \widetilde{\mathbf{R}}_2[\xi(e), \tau_h]\right]$$

We suppose that the covariates $\widetilde{\mathbf{R}}_2[\xi, \tau_h]$ are stored separately, in a secondary table indexed by $\{\xi$-values $\times$ time-breakpoints$\}$. If the key variable is a much coarser breakdown than individual (e.g. city), then preparing the secondary table is correspondingly easier. The secondary table properties (e.g. filename, key-variable, etc.) are given to the program in the control file, as described in the manual.

The algorithms for using a secondary table are essentially the same as Algorithm 2 above; as already mentioned, the conceptual data matrix $\mathbf{M}$ is the same in either case. To handle a secondary table, we construct indexing structures that allow stepping through the primary and secondary rows in the same order as the rows of $\mathbf{M}$. In effect we apply Algorithm 2, building the rows of $\mathbf{M}$ on the fly.

Indexing Algorithms

There are many other algorithms required for handling the data structures used by the program: we must compile lists of event times, compile delete-lists for each event-time, construct the cluster-tree, and other similar tasks. These indexing algorithms are sometimes complicated, but they raise no conceptual issues, so we will not give them here.

Standard Errors

Preliminaries

As already described, the regression coefficients α and β can be found by an iteration of the form

$$\mathbf{S}(\gamma_{old})(\gamma_{new} - \gamma_{old}) = -\psi(\gamma_{old})$$

Since the α-vector can be large, the equation is impractical in this form, and we instead use

$$K\beta = q$$

where K is the Schur complement of $\mathbf{S}$ with respect to α. The standard errors of the γ-coefficients (composite of α and β) are the diagonal entries of the inverse of the final converged value of $\mathbf{S}$, and it is easy to show that the standard errors of β alone are the diagonal entries of the inverse of K. For the iterations, we approximate K by $\hat{K}$ as already described, but $\hat{K}$ is not accurate enough to use for standard errors. So once, after convergence, we have to compute the exact K itself. This is a somewhat difficult

computation, because with the size of problems we are trying to handle, the matrices may be too big to hold in memory.

With the notation introduced in the first few sections, in particular the matrices $\mathbf{X}$, $\mathbf{E}$, $\mathbf{R}$, $\mathbf{S}$, A, B, Q, D, etc., we note from section "Estimation of β" that

$$\mathbf{S} = \widetilde{\mathbf{X}}^{\mathbf{T}} A \operatorname{var}(Y)^{-1} A \widetilde{\mathbf{X}}$$

and

$$\operatorname{var}(Y)^{-1} = A^{-1} - A^{-1}B(D^{-1} + Q)^{-1}B^{\mathbf{T}}A^{-1}$$

so

$$A \operatorname{var}(Y)^{-1} A = A - B(D^{-1} + Q)^{-1}B^{\mathbf{T}}$$

It follows, then, that

$$\begin{aligned}
\mathbf{S} &= \widetilde{\mathbf{X}}^{\mathbf{T}}[A - B(D^{-1} + Q)^{-1}B^{\mathbf{T}}]\widetilde{\mathbf{X}} \\
&= \begin{bmatrix} \widetilde{\mathbf{E}}^{\mathbf{T}}(A - B(D^{-1} + Q)^{-1}B^{\mathbf{T}})\widetilde{\mathbf{E}} & \widetilde{\mathbf{E}}^{\mathbf{T}}(A - B(D^{-1} + Q)^{-1}B^{\mathbf{T}})\widetilde{\mathbf{R}} \\ \widetilde{\mathbf{R}}^{\mathbf{T}}(A - B(D^{-1} + Q)^{-1}B^{\mathbf{T}})\widetilde{\mathbf{X}} & \widetilde{\mathbf{R}}^{\mathbf{T}}(A - B(D^{-1} + Q)^{-1}B^{\mathbf{T}})\widetilde{\mathbf{R}} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{S}_{\alpha}^{\alpha} & \mathbf{S}_{\beta}^{\alpha} \\ \mathbf{S}_{\alpha}^{\beta} & \mathbf{S}_{\beta}^{\beta} \end{bmatrix}, \text{ say.}
\end{aligned}$$

Our goal, then, is to compute the Schur complement

$$K = \mathbf{S}_{\beta}^{\beta} - \mathbf{S}_{\alpha}^{\beta}(\mathbf{S}_{\alpha}^{\alpha})^{-1}\mathbf{S}_{\beta}^{\alpha} \tag{27}$$

Since the coefficient vector β can be expected to be of relatively relatively low dimension (≤ 100, say) it is generally feasible to store the matrices $\mathbf{S}_{\beta}^{\alpha} = (\mathbf{S}_{\alpha}^{\beta})^{\mathbf{T}}$, $\mathbf{S}_{\beta}^{\beta}$ and K, and to invert K, but in large problems storing and inverting $\mathbf{S}_{\alpha}^{\alpha}$ is out of the question on most machines. We must find a relatively efficient way of computing $\mathbf{S}_{\alpha}^{\beta}(\mathbf{S}_{\alpha}^{\alpha})^{-1}\mathbf{S}_{\beta}^{\alpha}$ "out of core".

The Basic Algorithm

We note that

$$\begin{aligned}\mathbf{S}_\alpha^\alpha &= \widetilde{\mathbf{E}}^{\mathbf{T}}(A - B(D^{-1}+Q)^{-1}B^{\mathbf{T}})\widetilde{\mathbf{E}} \\ &= \widetilde{\mathbf{E}}^{\mathbf{T}}A\widetilde{\mathbf{E}} - \widetilde{\mathbf{E}}^{\mathbf{T}}B(D^{-1}+Q)^{-1}B^{\mathbf{T}}\widetilde{\mathbf{E}} \\ &= H - Z, \text{ say,}\end{aligned}$$

and $H = \widetilde{\mathbf{E}}^{\mathbf{T}}A\widetilde{\mathbf{E}}$ is a diagonal matrix, since A is. We concentrate, then, on

$$Z = \widetilde{\mathbf{E}}^{\mathbf{T}}B(D^{-1}+Q)^{-1}B^{\mathbf{T}}\widetilde{\mathbf{E}}.$$

Now suppose that we can factor the BLUP matrix $(D^{-1}+Q)^{-1}$ in some way:

$$(D^{-1}+Q)^{-1} = FG^{\mathbf{T}}$$

for some suitable choice of matrices F and G. One possibility is the Cholesky factorization, in which case $F = G$, but we leave open for now the particular factorization chosen: it may depend on the random effects covariance model used. With this factorization, the matrix Z can be written

$$Z = \widetilde{\mathbf{E}}^{\mathbf{T}}BFG^{\mathbf{T}}B^{\mathbf{T}}\widetilde{\mathbf{E}} = NM^{\mathbf{T}}, \text{ say,}$$

and if $F = G$, then $N = M$. By the Sherman-Morrison-Woodbury formula,

$$\begin{aligned}(\mathbf{S}_\alpha^\alpha)^{-1} &= (H-Z)^{-1} = (H - NM^{\mathbf{T}})^{-1} \\ &= H^{-1} + H^{-1}N(I - M^{\mathbf{T}}H^{-1}N)^{-1}M^{\mathbf{T}}H^{-1}\end{aligned} \tag{28}$$

The matrices N and M are of dimension "α by leaves", i.e. the rows are indexed by pairs sh of strata s and stratum event-times τ_h, and columns are indexed by leaf-clusters. The rows of N and M corresponding to stratum-time pair sh are:

$$N^{sh} = (\widetilde{\mathbf{E}}^{\mathbf{T}} BF)^{sh} = e^{\alpha^{sh}} \sum_{k \in R_{sh}} \exp(\mathbf{R}^k \beta) F^{r^k}$$
$$M^{sh} = (\widetilde{\mathbf{E}}^{\mathbf{T}} BG)^{sh} = e^{\alpha^{sh}} \sum_{k \in R_{sh}} \exp(\mathbf{R}^k \beta) G^{r^k}$$

where F^r and G^r are the rows of F and G corresponding to leaf-cluster r. Assuming that rows of F and G are available as needed, these forms can be computed by a variant of Algorithm 2 above: for a given stratum s, that algorithm steps through the event-times τ_h belonging to *s*, and on pass *h*, the rows N^{sh} and M^{sh} are produced, and immediately written to a binary file. On the same execution of Algorithm 2, we produce the other quantities needed, in particular $\widetilde{\mathbf{R}}^{\mathbf{T}} A \widetilde{\mathbf{R}}$, H, $B^{\mathbf{T}} \widetilde{\mathbf{R}}$, and $\widetilde{\mathbf{E}}^{\mathbf{T}} \widetilde{\mathbf{R}}$. The factorization $(D^{-1} + Q)^{-1} = FG^{\mathbf{T}}$ and the other computations involving the BLUP matrix are carried out by the modules of the covariance models, in whatever way is most efficient for the particular model. Having these, we can form the matrices $\mathbf{S}_\beta^\beta$, $\mathbf{S}_\beta^\alpha$ and $M^{\mathbf{T}} H^{-1} N$. Most of these require matrix products with N or M, or both. We do this by reading the rows of N and M back in sequence, and forming and accumulating rank-1 matrices with the individual rows, to produce the full matrix product. Reading and writing binary files are fast operations, but the overall process is nevertheless slow. Still, it is fast enough to be just feasible with large problems.

The matrix $(I - M^{\mathbf{T}} H^{-1} N)^{-1}$

As mentioned, the matrix $M^{\mathbf{T}} H^{-1} N$ can be formed sequentially by reading the rows of N and *M*. It is $N_{leaf} \times N_{leaf}$, which is not a problem if N_{leaf} is moderate. In some problems N_{leaf} can be large, however, and storing and inverting $I - M^{\mathbf{T}} H^{-1} N$ becomes

difficult or impossible. Instead we approximate the product $(I - M^{\mathtt{T}} H^{-1} N)^{-1} M^{\mathtt{T}} \mathbf{S}_{\beta}^{\alpha}$ by solving the matrix equation

$$(I - M^{\mathtt{T}} H^{-1} N) X = M^{\mathtt{T}}$$

for X, using the GMRES method of Saad and Schultz (1986) or Saad (2003). This method requires only that we be able to form $(I - M^{\mathtt{T}} H^{-1} N) v$, for given vectors v, and this can be done as outlined above, by successively reading in the rows of N and M, and forming dot products. The result of the process is the matrix $\mathbf{S}_{\alpha}^{\beta} (I - M^{\mathtt{T}} H^{-1} N)^{-1} M^{\mathtt{T}} \mathbf{S}_{\beta}^{\alpha}$, formed without ever storing $M^{\mathtt{T}} H^{-1} N$. However, GMRES is an iterative algorithm, and the issue arises of convergence and accuracy. It can be shown that $N(I - M^{\mathtt{T}} H^{-1} N)^{-1} M^{\mathtt{T}}$ is symmetric positive definite, and so we should expect $I - M^{\mathtt{T}} H^{-1} N$ to be reasonably well-conditioned. We have found in tests that only a few GMRES iterations are required to give accurate standard errors.