

Generalized Relational Event Models

- ◆ `rem.dyad()` is excellent for dyadic data
- ◆ The relational event framework is more general than the dyadic case
 - ◆ Proportional hazards for egocentric events, multiple event types, multiple event histories
- ◆ The `relevent` package contains methods for estimating general relational event models:
 - ◆ `rem()` is the function for fitting general models

Key differences between `rem()` and `rem.dyad()`

	<code>rem.dyad()</code>	<code>rem()</code>
Data structure	List of sender-receiver dyads, single event history	List of event types, multiple event histories (with optional support constraints)
Ordinal likelihood event order	User-supplied	Inferred from data structure
Optimization	<code>optim()</code>	<code>trust()</code>
Number of actors	User-supplied	Inferred from data structure
Sufficient statistics	Fixed-set, function-supplied	User-supplied

Key differences between `rem()` and `rem.dyad()`

	<code>rem.dyad()</code>	<code>rem()</code>
Data structure	List of sender-receiver dyads, single event history	List of event types, multiple event histories (with optional support constraints)
Ordinal likelihood event order	User-supplied	Inferred from data structure
Optimization	<code>optim()</code>	<code>trust()</code>
Number of actors	User-supplied	Inferred from data structure
Sufficient statistics	Fixed-set, function-supplied	User-supplied

User-supplied sufficient statistics

Each element of $g(y)$ needs to be constructed and supplied by the user. For each event-history, this will consist of an 3-D array called a statslist.

- ◆ i, j, k dimensions
 - ◆ i^{th} element is the order of the event
 - ◆ j^{th} element is the event type
 - ◆ k^{th} element is the model statistic

```
, , play
  play eat ran stay
1      1   0   0   0
2      1   0   0   0
3      1   0   0   0
, , eat
  play eat ran stay
1      0   1   0   0
2      0   1   0   0
3      0   1   0   0
, , stay
  play eat ran stay
1      0   0   0   1
2      0   0   0   1
3      0   0   0   1
```

User-supplied sufficient statistics

- ♦ Can be cumbersome to compute, construct
 - ♦ Especially with many event types, actors, and complex sequence statistics
- ♦ Use the *informR* companion package to relevent
 - ♦ Version 1.0-5 as of this conference
 - ♦ Available on cran
 - ♦ See Marcum & Butts (2015) in JSS



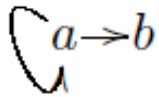
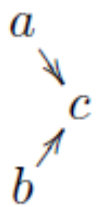
The *informR* package for R

- ◆ Aids in the construction of statslists for use by `rem()`
 - ◆ By automating the building and amending of complex arrays
- ◆ Methods for covariates on events and actors (limited)
- ◆ Used with `for` for both ordinal and interval data
- ◆ A simple event-type token system to specify statistics: “ab” for “event a → event b”

S-form notation

- ◆ **Sequence form (s-form for short) of model statistics**
 - ◆ **Let each s-form consist of two-parts**
 - ◆ **A prefix: The event or series of events that lead to a focal event to be predicted by the sufficient statistic.**
 - ◆ **A suffix: The event to be predicted by the sufficient statistic (preceded, of course, by the prefix).**
 - ◆ **$a \rightarrow b \rightarrow c$, event sequence $a \rightarrow b$ is the prefix, event c is the suffix**
- ◆ **See ?informR documentation for more details**

S-form notation

S-form	Regex	Definition
$a \rightarrow a$	aa	inertial term: S-form of the type “event a predicts event a ”
$a \rightarrow b$	ab	basic digram transition term: S-form of the type “event a predicts event b ”
 $a \rightarrow b$	$a + b$	transition term with persistence: S-form of the type “some series of events a predicts event b ”
$a \rightarrow b \rightarrow c$	abc	basic trigram transition term: S-form of the type “event a followed by event b predicts event c ”
	$(a b)c$	transition term with divergence: S-form of the type “event a OR event b predicts event c ”

Marcum & Butts (2015)

***informR*: limitations and future functionality**

Limitations

- ◆ Number of event-types limited to 52 ($26*2$)
- ◆ `glb.sformlist()` is slow when number of event-types X number of statistics is large
- ◆ Dyadic event structure not natively supported (but tools still useful)

Upcoming Features

- ◆ Arbitrary number of event-types
- ◆ Improved covariate routines
- ◆ Imputation of ``missing'' events (types and times)
- ◆ support constraint list functions

informR: limitations and future functionality

Limitations

Some new features are already implemented in development code.

Email:

chris.marcum@nih.gov

for details

Upcoming Features

- ◆ **Arbitrary number of event-types**
- ◆ Improved covariate routines
- ◆ **Imputation of "missing" events (types and times)**
- ◆ support constraint list functions

Examples in R

