Essex Summer School in Social Science Data Analysis
Panel Data Analysis for Comparative Research

Panel Data with Binary Dependent Variables

Christopher Adolph

Department of Political Science
and
Center for Statistics and the Social Sciences
University of Washington, Seattle

Event history analysis

A large field (and an Essex summer course by Alejandro Quiroz Flores) considers event history

An event history dataset records, for each of N units and T periods, whether an event happened (1) or not (0) in unit i and period t

Event history data are simply categorical time series cross-section data

Most commonly, there are only two possibilities:
war or peace, government failure or continuation, life or death

There are event history models for multiple unordered outcomes, however (e.g., war, peace, alliance), known as competing risk models

If you have a panel of ordered categorical time series, look into mixed autoregressive ordered probit in the `maop` package in `R` $\rightarrow$ *seems hard to find at the moment*

You could also consider models for lagged latent dependent variables generally, and hidden Markov models. Some of these models are best handled with MCMC.

Survival models

We will talk only about binary time series cross-section,
and only as it most closely links to our core panel data topics

But first, we need to understand some essential duration concepts

Running example:
Suppose we are studying how long it takes for a government to fall

Survival models: Key concepts

Let the *duration*, D_i, be the time elapsed from the start of the government to the moment it collapses for country i

Let the *survival function*, S_{it}, indicate the probability a government lives past a given point in time:

$$S_{it} = \mathrm{P}(D_i > t)$$

Define the *lifetime distribution function*, F_{it}, as the probability a government has died by time t:

$$F_{it} = 1 - S_{it}$$

Define *event density*, f_{it}, as the probability of government failure at t precisely:

$$f_{it} = \frac{dF_{it}}{dt}$$

Define the *hazard rate*, h_{it} as the probability of failure at t precisely *given* survival to time t:

$$h_{it}dt = \mathrm{P}(t \leq D_i < t + dt) = \frac{f_{it}}{S_{it}}$$

Cox Proportional Hazards

The hazard, or the chance that a currently running process is about to fail,
is what we want to model as a function of covariates

Most important approach: Cox Proportional Hazards Model

CPH assumes there is some baseline hazard function, h_{ot}, which varies over time

The shape of the baseline hazard may be highly complex,
reflecting numerous idiosyncracies in the "usual" course of a process

Cox Proportional Hazards

We allow for those idiosyncracies,
and estimate the shape of the baseline hazard non-parametrically

Then we simply let the hazard rate for any actual process be a multiple of the baseline hazard (estimated by maximum likelihood):

$$h_{it} = h_{0t}\exp(x_i\beta)$$

The upshot is that differences in x_i proportionally increase or lower the hazard, or probability of failure in an on-going process

To see this, consider governments, i and j, both at risk of failure

The relative probability of failure at t is given by:

$$\frac{h_{it}}{h_{jt}} = \frac{h_{0t}\exp(x_i\beta)}{h_{0t}\exp(x_j\beta)} = \frac{\exp(x_i\beta)}{\exp(x_j\beta)}$$

Cox Proportional Hazards

$$\frac{h_{it}}{h_{jt}} = \frac{h_{0t}\exp(x_i\beta)}{h_{0t}\exp(x_j\beta)} = \frac{\exp(x_i\beta)}{\exp(x_j\beta)}$$

Suppose $x_j = 0$ and $x_i = 1$. Then:

$$\frac{\exp(x_i\beta)}{\exp(x_j\beta)} = \frac{\exp(\beta \times 1)}{\exp(\beta \times 0)} = \frac{\exp(\beta)}{\exp(0)} = \exp(\beta)$$

This lets us easily interpret exponentiated Cox regression coefficients:

if $\exp(\hat{\beta}_1) = 0.25$ then $\uparrow x_1$ by 1 leads to $\downarrow \mathrm{P}(y)$ by 75% vs baseline

if $\exp(\hat{\beta}_1) = 1.80$ then $\uparrow x_1$ by 1 leads to $\uparrow \mathrm{P}(y)$ by 80% vs baseline

Cox Proportional Hazards

To estimate the Cox Proportional Hazards model in `R`, use the `survival` library commands:

```
# We need a vector of starting times, a vector of ending times,
# and whether an event has occurred by the ending time
duration <- Surv(start, stop, event)
res <- coxph(duration~x1+x2+x3)
```

The `survfit` command is also very helpful for predicting conditional survival curves

Note two things above:

1. The model accounts for observation that are "right-censored"
(no failure yet when "time is up")

Cox Proportional Hazards

To estimate the Cox Proportional Hazards model in `R`, use the `survival` library commands:

```
# We need a vector of starting times, a vector of ending times,
# and whether an event has occurred by the ending time
duration <- Surv(start, stop, event)
res <- coxph(duration~x1+x2+x3)
```

The `survfit` command is also very helpful for predicting conditional survival curves

Note two things above:

2. If covariates change over time within units, the model treats the different "phases" of a unit in which covariates are static as different observations, each of which is right-censored, except (perhaps) the final period

See Box-Steffensmeier and Jones' excellent introductory text for more on including time-varying covariates in Cox PH models

Event history models

If you know how to use event history models,
you already know how to model binary time series cross-sectional data

Well developed models with lots of useful tools—how I model these data

But if you don't know these models,
in some cases a simple modification of logit will suffice. . .

Binary Time Series Cross Section

Beck, Katz, and Tucker (1998, AJPS) offer some simple tricks
for turning ordinary logit into Cox proportional hazards model

Suppose we have a binary indicator of whether an event occurred in unit i at time t

We might suppose this event is a function of past events, covariates, and lags of covariates:

$$\mathrm{P}(y_{it}) = f(y_{i,t-1}, \ldots, y_{i,t-p}, x_{1it}, \ldots, x_{kit}, x_{1,i,t-1}, \ldots)$$

But we can't just stick this into linear regression—y_{it} is binary, so least squares is highly inefficient, and has biased standard errors (due to heteroskedasticity)

Nor can we just stick lags into logit—that trick only works for linear models

(Note also we might want the lag of the latent variable $y^*_{i,t-1}$,
not the lagged realization $y_{i,t-1}$)

BTSCS: What *not* to do

A common problem in IR, e.g., in the study of war onset among dyads

But also a problem for studies of policy adoption, lifecycle events, etc.

Years ago in IR, political scientists often gave up and estimated an ordinary logit:

$$\mathrm{P}(y_{it} = 1 | x_{i,t}) = \frac{1}{1 + \exp(-x_{it}\beta)}$$

This model is consistent but very inefficient, and with very biased standard errors

BTSCS *is* event history

Binary time series cross-section is just a discrete case of event history

Event history models (also called survival or duration analysis)
model the time until an event occurs for each of N cases

Usually, these models are in continuous time

BTSCS is just a discrete version, where the time periods are highly aggregated

Instead of knowing a period of peace lasted 4 years, 3 months, and 2 days,
we might just have four periods of peace, followed by a period of war

BTSCS as a proportional hazards model

The discrete version of a proportional hazard model is:

$$\mathrm{E}(h_t|x_{i,t}) = h_{0,t}\exp(x_{it}\beta)$$

The $\exp(x_{it}\beta)$ turns out to be our logit model,
but we are missing the baseline hazard $h_{0,t}$

For continuous time, this is a (potentially) very complex function of time

But for discrete time periods, it must be a simple step function

Any step function can be decomposed into a set of dummy variables,
one for each step

BKT note that for discrete baseline hazards

$$
\begin{aligned}
\mathrm{E}(h_t|x_{i,t}) &= h_{0,t}\exp(x_{it}\beta) \\
&= 1 - \exp(-\exp(x_{it}\beta + \kappa_{t-t0})) \\
&\approx \frac{1}{1+\exp(-x_{it}\beta - \kappa_{t-t0})} \qquad \text{for } h_t < 0.5
\end{aligned}
$$

κ_{t-t0} is a duration dummy indicating the number of periods since the last event:

t	y	duration	dummy	κ_1	κ_2	κ_3	κ_4
1	0	1	κ_1	1	0	0	0
2	0	2	κ_2	0	1	0	0
3	0	3	κ_3	0	0	1	0
4	1	4	κ_4	0	0	0	1
5	0	1	κ_1	1	0	0	0
6	0	2	κ_2	0	1	0	0
7	1	3	κ_3	0	0	1	0
8	1	1	κ_1	1	0	0	0
9	0	1	κ_1	1	0	0	0
10	0	2	κ_2	0	1	0	0

BTSCS with duration dummies

The simplest BTSCS model provided by BKT is this logit:

$$\mathrm{P}(y_{it} = 1|x_{it}) = \frac{1}{1 + \exp(-x_{it}\beta - \kappa_{t-t0})}$$

κ_{t-t0} is a duration dummy indicating the number of periods since the last event

To do this in `R`, we just need to create these dummies and put them in

```
glm(y~x+dummies, family=binomial)
```

BTSCS with smoothing splines

One problem with duration dummies is the difficulty in estimating rarely appearing durations (i.e., long ones).

It may be reasonable to assume the baseline hazard is smooth:

$$\mathrm{P}(y_{it} = 1|x_{it}) = \frac{1}{1 + \exp(-x_{it}\beta - \mathrm{smooth}(\kappa_{t-t0}))}$$

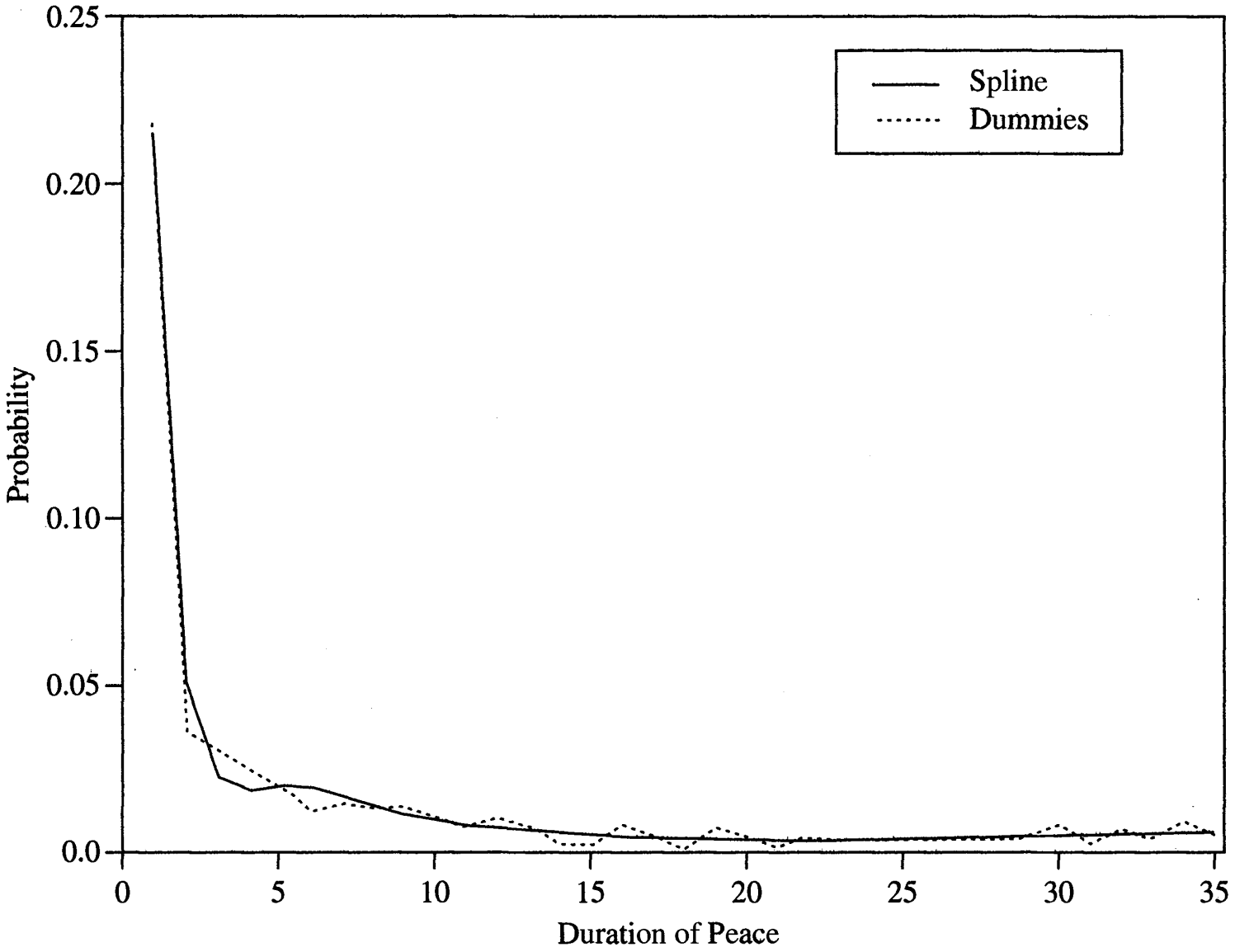
Then we need only estimate a smoother of the time series dummies,
which might have only three parameters or so

To apply a smoothing spline to the dummies, create the duration count `duration`, load the `mgcv` package and try

```
gam(y~x+s(duration), family=binomial)
```

Democratic peace example

Figure 2. Discrete Hazard of Dispute



Democratic peace example

	Ordinary Logit	Grouped Duration		
		Logit Dummy[a]	Logit Spline	Cloglog Dummy[b]
Variable	I	II	III	IV
Democracy	–0.50	–0.55	–0.54	–0.49
	(0.07)	(0.08)	(0.08)	(0.07)
Economic Growth	–2.23	–1.15	–1.15	–0.81
	(0.85)	(0.92)	(0.92)	(0.76)
Alliance	–0.82	–0.47	–0.47	–0.43
	(0.08)	(0.09)	(0.09)	(0.08)
Contiguous	1.31	0.70	0.69	0.55
	(0.08)	(0.09)	(0.09)	(0.08)
Capability Ratio	–0.31	–0.30	–0.30	–0.30
	(0.04)	(0.04)	(0.04)	(0.04)
Trade	–66.13	–12.67	–12.88	–12.50
	(13.44)	(10.50)	(10.51)	9.96
Constant	–3.29	–0.94	–0.96	–1.11
	(0.08)	(0.09)	(0.09)	(0.08)
Peace Years			–1.82	
			(0.11)	
Spline(1) [c]			–.24	
			(0.03)	
Spline(2) [c]			–.08	
			(0.01)	
Spline(3)[c]			–.01	
			(0.003)	
Log Likelihood	–3477.6	–2554.7	–2582.9	–2554.1
df	20983	20036	20979	20949
N=20990				

BTSCS with random effects

We might suppose that some peace among some dyads
is just randomly stronger or weaker that others

A large random intercept for a dyad would mean that dyad is more frail,
hence we call these random effects "frailties"

You can also add random effects for intercepts
to either the duration dummies or smoothing spline model

See `glmmML()` in the `glmmML` library to use frailties and duration dummies

See `gamm()` in the `mgcv()` library to use frailties and smoothing splines

BTSCS: things to remember

1. For observation i,
   the initial spell of peace may prexist the initial year of your dataset.

   If you know how long the first spell really lasted,
   you should include the appropriate duration dummies

2. If you have repeated events for a single unit,
   consider including a variable with the count of past events, to control for dynamics

3. This model cannot include unit-invariant contemporaneous shocks,
   as they will likely be too strongly correlated with the duration dummies

   By the same token, don't add period dummies

4. Finally, these models don't handle missing data at all well.
   (that is, how do you listwise delete without losing the whole case?)