A Scalable Empirical Model of Waiting Times

Harry J. Paarsch
Department of Economics
University of Central Florida

Orlando Machine Learning and Data Science Meetup
November 2018

Motivation

- Many phenomena in data science involve waiting until a particular event occurs.
- For example, ordered merchandise does not arrive immediately. Usually, it takes some time for transportation companies to deliver items from vendors to firms.
- Because of random lags in order arrivals, zero inventories can obtain along with their accompanying costs.
- Therefore, understanding the stochastic process for the arrival of purchased merchandise at firms is an important component of supply-chain management.

Summary

- I develop a flexible, discrete-time, semiparametric statistical model of waiting times that can be trained on data concerning intervals of different, user-defined lengths—for example, at the hourly, eight-hourly, daily, weekly, or monthly interval.
- The building block is the Bernoulli distribution.
- For example, let B denote a Bernoulli random variable: When B is zero the event has not yet occurred, whereas when B is one the event has occurred.

Geometric Distribution

- Consider a sequence of independent Bernoulli random variables $\{B_\tau\}_{\tau=1}^{t}$, each of which takes on the value zero with probability $(1-\pi)$, or one with probability π.
- How many periods must one wait until B equals one, that is, the event occurs?
- Well, it could be one or two or three, even more.
- In general, the waiting time T is a random variable having the following probability mass function (pmf):

$$p_T(t;\pi) = (1-\pi)^{t-1}\pi \quad t = 1, 2, \ldots$$

which is often referred to as the *geometric distribution*.

Continuous Waiting Times

- In the continuous case, the waiting time to some event can be represented by a positive random variable T that has probability density function (pdf) denoted $f_T(t)$ and cumulative distribution function (cdf) denoted $F_T(t)$ where

$$F_T(t) = \Pr(T \leq t) = \int_0^t f_T(z)\ \mathrm{d}z.$$

- In words, $F_T(t)$ represents the fraction of events that will obtain within the first t periods.
- The related fraction of events that have not obtained by t is

$$\Pr(T > t) = 1 - \Pr(T \leq t) = \int_t^\infty f_T(z)\ \mathrm{d}z = [1 - F_T(t)],$$

often referred to as the *survivor function* and denoted $S_T(t)$.

Duration Models

- In the statistics literature, the standard way to investigate waiting-time random variables is to use techniques from a subfield referred to by several different names, including *duration analysis*, *failure-time analysis*, and *survival analysis*. Below, I shall use duration analysis.
- In duration analysis, a number of different models can be employed to put structure on the probability law governing T, the pdf $f_T(t)$, but the three most common ones are the exponential, Weibull, and Gamma laws.

Three Laws

- The exponential law has pdf

$$f_T(t) = \lambda \exp(-\lambda t) \quad \lambda > 0,\ t > 0,$$

whereas the Weibull law has pdf

$$f_T(t) = \lambda p t^{p-1} \exp(-\lambda t^p) \quad \lambda > 0,\ p > 0,\ t > 0,$$

and the Gamma law has pdf

$$f_T(t) = \frac{\lambda^q}{\Gamma(q)} t^{q-1} \exp(-\lambda t) \quad \lambda > 0,\ q > 0,\ t > 0.$$

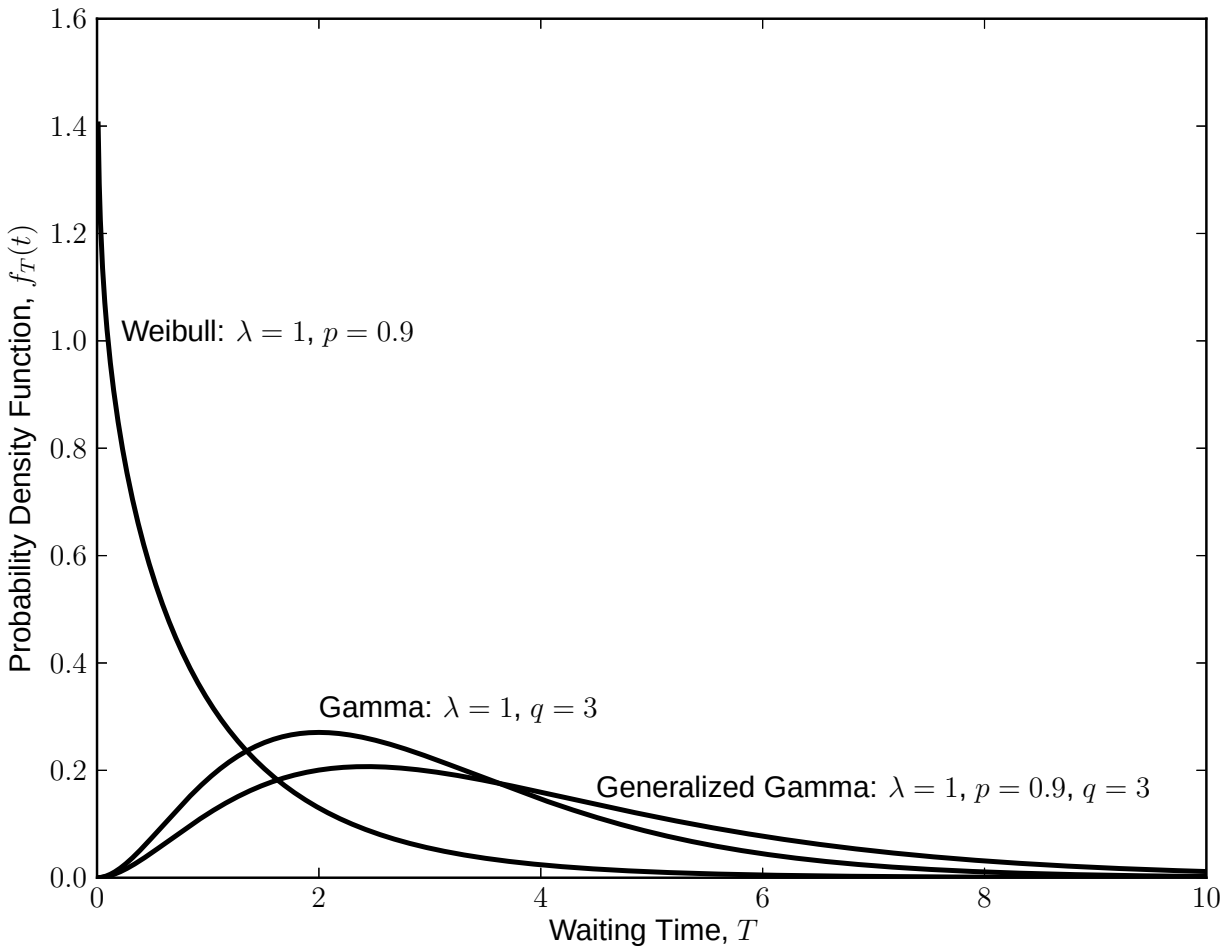
- Obviously, the Weibull and the Gamma laws are generalizations of the exponential law.

Encompassing Model

- Often, λ is referred to as the *hazard rate* of the exponential distribution.
- Sometimes, p is referred to as the *shape parameter* of the Weibull distribution.
- A model that encompasses all three laws is the generalized Gamma law whose pdf is

$$f_T(t) = \frac{\lambda^q p}{\Gamma(q/p)} t^{q-1} \exp(-\lambda^p t^p) \quad \lambda > 0,\ p > 0,\ q > 0,\ t > 0.$$

Generalized Gamma Densities



Hazard Rate

- Now, the conditional pdf of T, given that an event has not obtained by t, is

$$\frac{f_T(t)}{\Pr(T > t)} = \frac{f_T(t)}{[1 - \Pr(T \leq t)]} = \frac{f_T(t)}{[1 - F_T(t)]} = \frac{f_T(t)}{S_T(t)} \equiv h_T(t).$$

- The conditional pdf defined above is often referred to as the *hazard rate* or the *hazard function*.

Different Forms of Duration Dependence

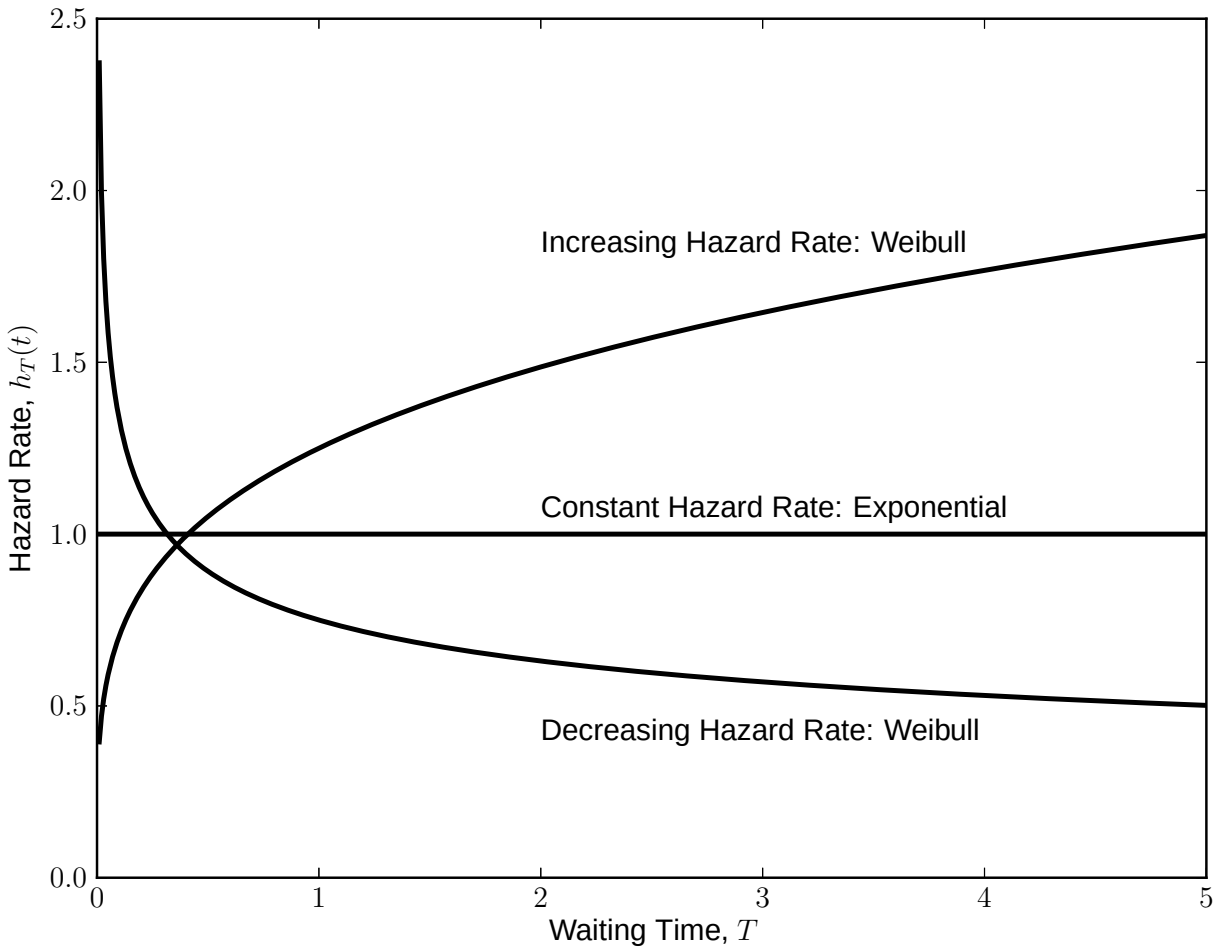
- Different combinations of the parameters (λ, p, q) translate into different conditional probability density functions of T.
- In the case of the exponential law, $p = q = 1$, so

$$h_T(t) = \lambda,$$

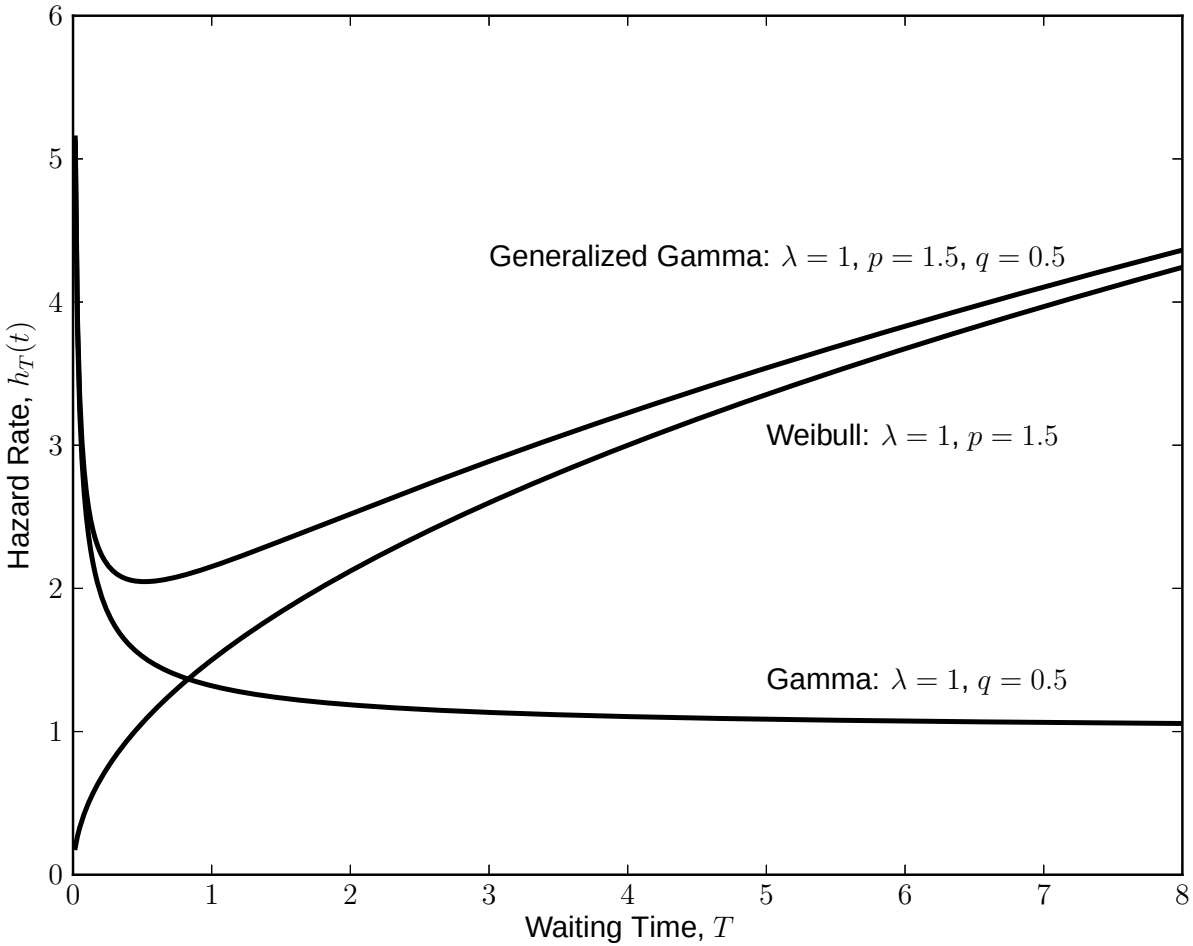
whereas for the Weibull law $q = 1$, so

$$h_T(t) = p\lambda t^{p-1}.$$

Exponential and Weibull Hazard Rates



Generalized Gamma Hazard Rates



Exponential and Geometric Random Variables

- The geometric distribution is the discrete-time analog of the exponential distribution.
- Like an exponentially-distributed random variable, a geometrically-distributed random variable has a constant hazard rate; that is, given that an event has not obtained in the first $(t-1)$ periods, the probability of its obtaining in period t is just π.
- The average waiting time for a geometric random variable is

$$\mathbb{E}(T) = \sum_{t=1}^{\infty} t p_T(t;\pi) = \frac{1}{\pi}.$$

Loosening the Constant Hazard Rate

As with the exponential law, one would like to loosen the constant hazard-rate assumption in several ways:

1. allow the hazard rate to vary with time-invariant features;
2. allow the hazard rate to vary with time, but perhaps not in the monotonic way admitted by the Weibull law or the restrictive U-shaped way admitted by some parametrizations of the generalized Gamma law;
3. allow the hazard rate to to vary with events that evolve over time, sometimes referred to as *time-varying covariates*.

Summary

- Including all of these requirements in a continuous-time model of duration can be challenging to do, mostly for computational reasons.
- The model I describe below is designed to be computationally tractable and, therefore, scalable, yet addresses (at least to the first order) each of the three requirements mentioned above.

Admitting Time-Invariant Features

- First, to introduce event-specific characteristics, I assume that a $(1 \times I)$ vector of features $\mathbf{u}$ exists for each observation and that these features influence the hazard rate π through a conformable $(I \times 1)$ vector of unknown parameters $\boldsymbol{\beta}$.
- That is, for observation a,

$$\pi_a = h(\mathbf{u}_a\boldsymbol{\beta}) \equiv \frac{\exp(\mathbf{u}_a\boldsymbol{\beta})}{1 + \exp(\mathbf{u}_a\boldsymbol{\beta})}.$$

- I make the logistic assumption for analytical and computational tractability.

Admitting a Time-Varying Hazard Rate

- Next, to introduce a time-varying hazard rate, which is sometimes referred to as *duration dependence*, I assume that

$$\pi_{a,t} = h(\mathbf{u}_a\boldsymbol{\beta} + \mathbf{v}_{a,t}\boldsymbol{\gamma}) \equiv \frac{\exp(\mathbf{u}_a\boldsymbol{\beta} + \gamma_t)}{1 + \exp(\mathbf{u}_a\boldsymbol{\beta} + \gamma_t)}$$
$$\equiv \frac{\exp(\mathbf{u}_a\boldsymbol{\beta} + \mathbf{v}_{a,t}\boldsymbol{\gamma})}{1 + \exp(\mathbf{u}_a\boldsymbol{\beta} + \mathbf{v}_{a,t}\boldsymbol{\gamma})}$$

  where $\mathbf{v}$ is a $(1 \times J)$ vector that has zeros for all elements except the t^{th} one, which equals one, and $\boldsymbol{\gamma}$ is a conformable $(J \times 1)$ vector of unknown parameters.

Admitting Time-Varying Features

- Finally, to introduce time-varying, event-specific characteristics, I assume that

$$\pi_{a,t} = h(\mathbf{x}_{a,t}\boldsymbol{\theta}) \equiv \frac{\exp(\mathbf{u}_a\boldsymbol{\beta} + \mathbf{v}_{a,t}\boldsymbol{\gamma} + \mathbf{w}_{a,t}\boldsymbol{\delta})}{1 + \exp(\mathbf{u}_a\boldsymbol{\beta} + \mathbf{v}_{a,t}\boldsymbol{\gamma} + \mathbf{w}_{a,t}\boldsymbol{\delta})}$$
$$\equiv \frac{\exp(\mathbf{x}_{a,t}\boldsymbol{\theta})}{1 + \exp(\mathbf{x}_{a,t}\boldsymbol{\theta})}$$

where $\mathbf{w}$ is a $(1 \times K)$ vector of features that can vary with time, while $\boldsymbol{\delta}$ is a conformable $(K \times 1)$ vector of unknown parameters.

Creating Features

- As with any empirical exercise in data science, the power of the analysis is largely determined by the features used.
- In this case, three different kinds of features exist, which I have denoted by $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ above.
- The elements of $\mathbf{u}$ are static; for the most part, these will be qualitative variables, for example indicator variables for order type or product line as in the case of order arrivals.
- On the other hand, the elements of $\mathbf{v}$ are generated features; these elements are binary outcomes, which vary with the event duration.
- The elements of $\mathbf{w}$ are time-varying, examples of which could include signals from a vendor and so forth.

Data Example

- Consider two examples: suppose that the duration until the first event is five, whereas the duration until the second event is eight.
- Suppose that J is eight.
- Let $\mathbf{u}$ be totally general.
- Let $\mathbf{w}$ be a scalar w which is incremented by one from some initial value, but that it can change from time to time, for whatever reason,

Table of Feature Variables for the Stylized Example

a	t_a	$y_{a,t}$	$\mathbf{u}$	$v_{a,1}$	$v_{a,2}$	$v_{a,3}$	$v_{a,4}$	$v_{a,5}$	$v_{a,6}$	$v_{a,7}$	$v_{a,8}$	$w_{a,t}$
1	1	0	$\mathbf{u}_1$	1	0	0	0	0	0	0	0	-6
1	2	0	$\mathbf{u}_1$	0	1	0	0	0	0	0	0	-5
1	3	0	$\mathbf{u}_1$	0	0	1	0	0	0	0	0	-4
1	4	0	$\mathbf{u}_1$	0	0	0	1	0	0	0	0	-3
1	5	1	$\mathbf{u}_1$	0	0	0	0	1	0	0	0	-2
2	1	0	$\mathbf{u}_2$	1	0	0	0	0	0	0	0	-4
2	2	0	$\mathbf{u}_2$	0	1	0	0	0	0	0	0	-3
2	3	0	$\mathbf{u}_2$	0	0	1	0	0	0	0	0	-2
2	4	0	$\mathbf{u}_2$	0	0	0	1	0	0	0	0	-3
2	5	0	$\mathbf{u}_2$	0	0	0	0	1	0	0	0	-2
2	6	0	$\mathbf{u}_2$	0	0	0	0	0	1	0	0	-1
2	7	0	$\mathbf{u}_2$	0	0	0	0	0	0	1	0	-0
2	8	0	$\mathbf{u}_2$	0	0	0	0	0	0	0	1	1

Constructing the Loss Function

- Consider an example concerning event a which has duration t_a and features $\mathbf{x}_{a,t}$.
- For the time being, ignore a- and t-specific information and focus on just one index n.
- Denote the label for example n by y_n where this label will be either zero or one.
- For any particular event a, having duration t_a, $(t_a - 1)$ of the y_n labels are zero, while the last one is one.

Logit Building Block

- Under the logistic assumption,

$$\Pr(y_n = 1|\mathbf{x}_n\boldsymbol{\theta}) = h(\mathbf{x}_n\boldsymbol{\theta}) = \frac{\exp(\mathbf{x}_n\boldsymbol{\theta})}{1 + \exp(\mathbf{x}_n\boldsymbol{\theta})},$$

  whereas

$$\Pr(y_n = 0|\mathbf{x}_n\boldsymbol{\theta}) = 1 - h(\mathbf{x}_n\boldsymbol{\theta}) = \frac{1}{1 + \exp(\mathbf{x}_n\boldsymbol{\theta})}.$$

- For example n, the loss function is often referred to as the *cross-entropy* function: It is simply the negative of the contribution to the logarithm of the likelihood function for example n.

Cross-Entropy Loss Function

- In symbols, the loss function for example n is

$$\begin{aligned}\ell(y_n, \mathbf{x}_n\boldsymbol{\theta}) &= -\, y_n \log[h(\mathbf{x}_n\boldsymbol{\theta})] - (1 - y_n) \log[1 - h(\mathbf{x}_n\boldsymbol{\theta})] \\ &= -\, y_n \mathbf{x}_n\boldsymbol{\theta} + y_n \log[1 + \exp(\mathbf{x}_n\boldsymbol{\theta})] + \\ &\qquad \log[1 + \exp(\mathbf{x}_n\boldsymbol{\theta})] - y_n \log[1 + \exp(\mathbf{x}_n\boldsymbol{\theta})] \\ &= -\, y_n \mathbf{x}_n\boldsymbol{\theta} + \log[1 + \exp(\mathbf{x}_n\boldsymbol{\theta})].\end{aligned}$$

Sample-Averaged Cross-Entropy Function

- The sample-averaged loss function when $N = \sum_{a=1}^{A} t_a$ examples exist is

$$L(\boldsymbol{y}, \mathbf{X}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \mathbf{x}_n \boldsymbol{\theta})$$
$$= \frac{1}{N} \sum_{n=1}^{N} \{-y_n \mathbf{x}_n \boldsymbol{\theta} + \log[1 + \exp(\mathbf{x}_n \boldsymbol{\theta})]\}$$

  where $\boldsymbol{y}$ is an $(N \times 1)$ vector that collects the labels of the N examples, while $\mathbf{X}$ is the $[N \times (I + J + K)]$ matrix concerning the features.

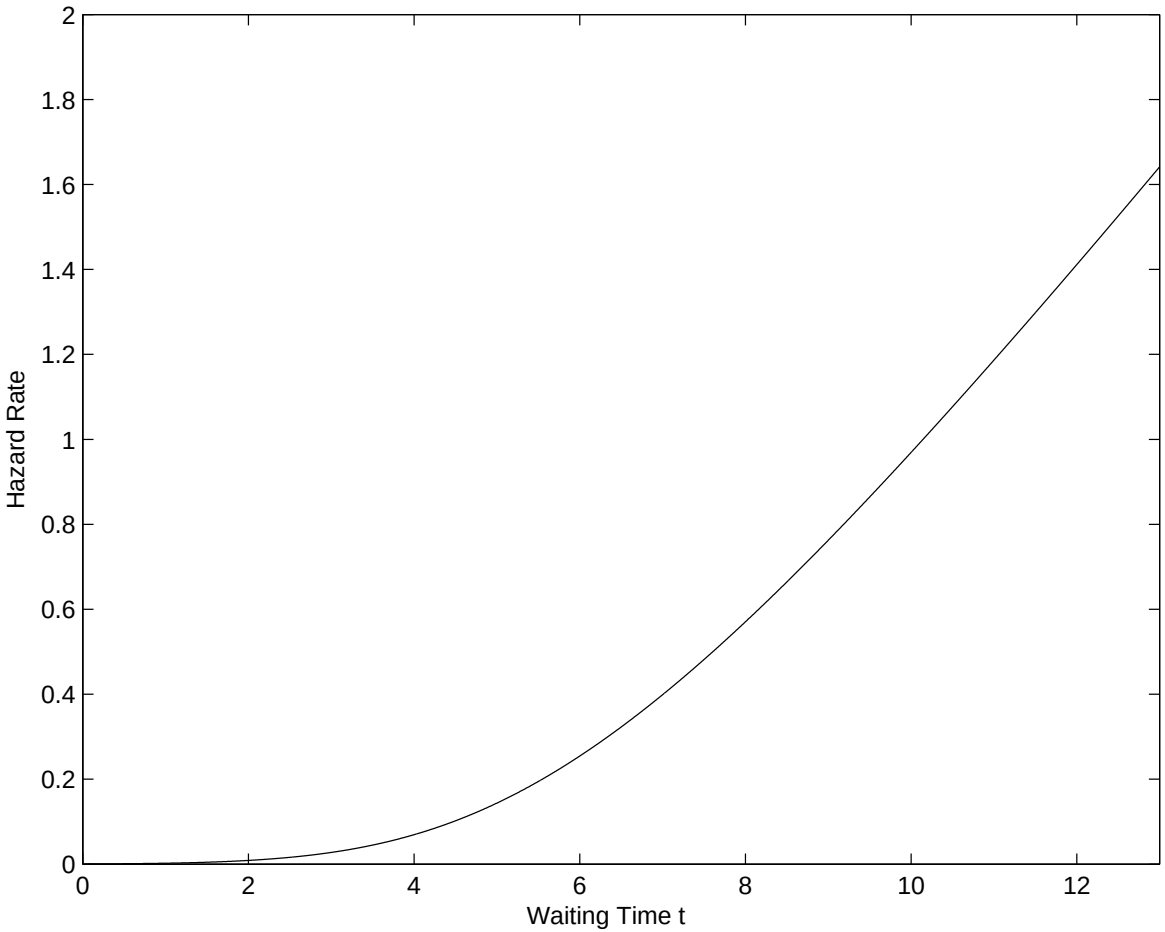
Comments and Replies

- Perhaps the most frequent comment about this framework is that it appears complicated, perhaps not even scalable.
- Why not just regress T_a on the vector $\mathbf{u}_a$? Better still, why not estimate a random forest?
- Two points:
  1. Linear regression and random forests are models typically used for the mean: This is a model of how the conditional probability evolves. One can derive a model of the mean from this model, but many different models of the conditional probability can give rise to the same random forest model.
  2. Neither a linear regression model nor a random forest model can admit the progession of time in a tractable, scalable way.
- In short linear regression and random forest models are inappropriate for the phenomenon being studied.

Additional Remarks

- Perhaps the most important phenomenon in a duration model is the effect that the passage of time has on the probability of the event's obtaining. In linear regression and random forest models, no natural way exists to measure how the odds of an event's obtaining change with the passage of time.
- In this duration model, the hazard rate is simply $h(\mathbf{x}_{a,t}\boldsymbol{\theta})$.
- As an aside, the linear regression model with Gaussian errors would have the following hazard rate:

Gaussian Hazard Rate



Important Point to Note

- To wit, if an event has not yet obtained today, then it is more likely to obtain tomorrow; if it does not obtain tomorrow, then it is even more likely to obtain the day after that, and so forth.

Relationship to Continuous-Time Model

- What primitives does the logistic model recover?
- Consider the Cox proportional hazard rate (CPHR) model, the most commonly used duration model. In that case,

$$h_{T|\mathbf{U}}(t|\mathbf{u}) = \exp(\mathbf{u}\boldsymbol{\beta})h_0(t)$$

  where $h_0(t)$ is the baseline hazard function.
- Now,

$$S_0(t) = \exp\left[-\int_0^t h_0(z)\ \mathrm{d}z\right] = [1 - F_0(t)],$$

  so

$$S_{T|\mathbf{U}}(t|\mathbf{u}) = S_0(t)^{\exp(\mathbf{u}\boldsymbol{\beta})} = [1 - F_0(t)]^{\exp(\mathbf{u}\boldsymbol{\beta})}.$$

Relationship to Continuous-Time Model (continued)

- Consequently,

$$F_{T|\boldsymbol{X}}(t|\boldsymbol{x}) = 1 - [1 - F_0(t)]^{\exp(\boldsymbol{x}\boldsymbol{\theta})}.$$

and

$$\begin{aligned} f_{T|\boldsymbol{X}}(t|\boldsymbol{x}) &= h_{T|\boldsymbol{X}}(t|\boldsymbol{x}) S_{T|\boldsymbol{X}}(t|\boldsymbol{x}) \\ &= \exp(\boldsymbol{x}\boldsymbol{\theta}) h_0(t) S_0(t)^{\exp(\boldsymbol{x}\boldsymbol{\theta})} \\ &= \exp(\boldsymbol{x}\boldsymbol{\theta}) h_0(t) \left\{ \exp\left[-\int_0^t h_0(z)\, \mathrm{d}z \right] \right\}^{\exp(\boldsymbol{x}\boldsymbol{\theta})}. \end{aligned}$$

Relationship to Continuous-Time Model (continued)

- Introduce

$$H_0(t) = \int_0^t h_0(z)\ \mathrm{d}z$$

which is often referred to as the *integrated hazard rate* as well as

$$\Lambda_0(t) = \log[H_0(t)].$$

- Now, under the CPHR model,

$$\Lambda_0(t) - \mathbf{u}\boldsymbol{\beta} = \log[H_0(t)] - \mathbf{u}\boldsymbol{\beta} = E.$$

where E is a random variable.

Relationship to Continuous-Time Model (continued)

- The probability of an event's obtaining in the interval $(t_{\text{lower}}, t_{\text{upper}}]$ is then

$$\int_{\Lambda_0(t_{\text{lower}})-\mathbf{u}_a\boldsymbol{\beta}}^{\Lambda_0(t_{\text{upper}})-\mathbf{u}_a\boldsymbol{\beta}} f_E(e)\ \mathrm{d}e.$$

- In short, if t_{upper} is an integer τ and t_{lower} is $(\tau - 1)$, then

$$\Pr\{T_a \in (\tau - 1, \tau]\} = h(\mathbf{u}_a, \mathbf{v}_{a,\tau}).$$

Relationship to Continuous-Time Model (continued)

- The logarithm of the likelihood function can be written as

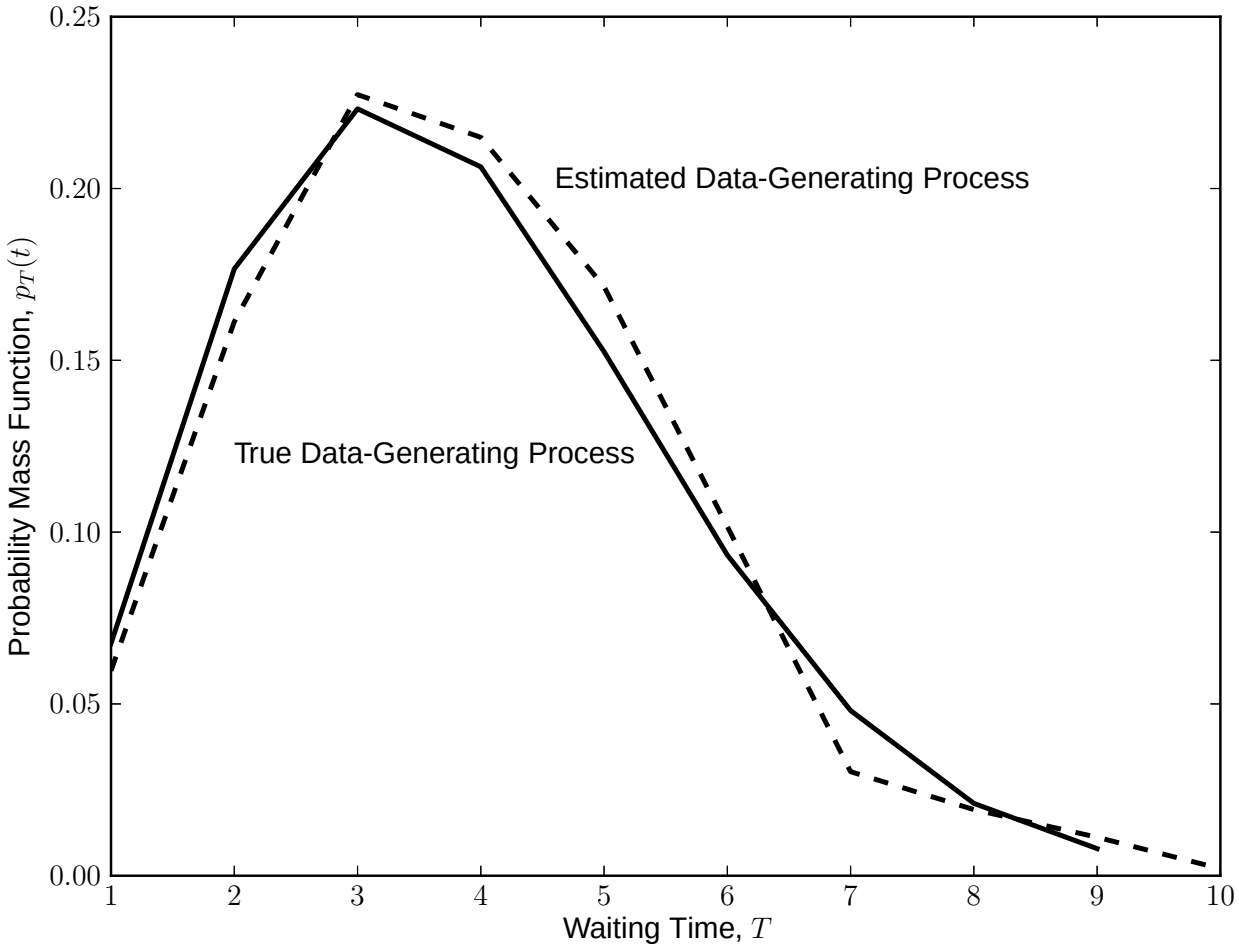
$$\mathcal{L}\left[\boldsymbol{\beta}, \Lambda_0(1), \ldots, \Lambda_0(J)\right] = \sum_{a=1}^{A} \sum_{\tau=1}^{\texttt{ceil}(t_a)} y_{a,\tau} \log \left[\int_{\Lambda_0(\tau-1)-\mathbf{u}_a\boldsymbol{\beta}}^{\Lambda_0(\tau)-\mathbf{u}_a\boldsymbol{\beta}} f_E(e)\, \mathrm{d}e \right]$$

where $\texttt{ceil}(t_a)$ is the ceiling of the float t_a.

Accuracy

- Suppose
$$f_T(t; \lambda, p) = p\lambda t^{p-1} \exp(-\lambda t^p)$$
where $\lambda = 0.07$ and $p = 2$, and U distributed uniformly on the interval $[0, 1]$.
- I used simulation to generate a data set.
- In this data set are $1,000$ events, which resulted in $N = 2,534$ binary observations.
- How well does the logistic model do at recovering the primitive data-generating process?

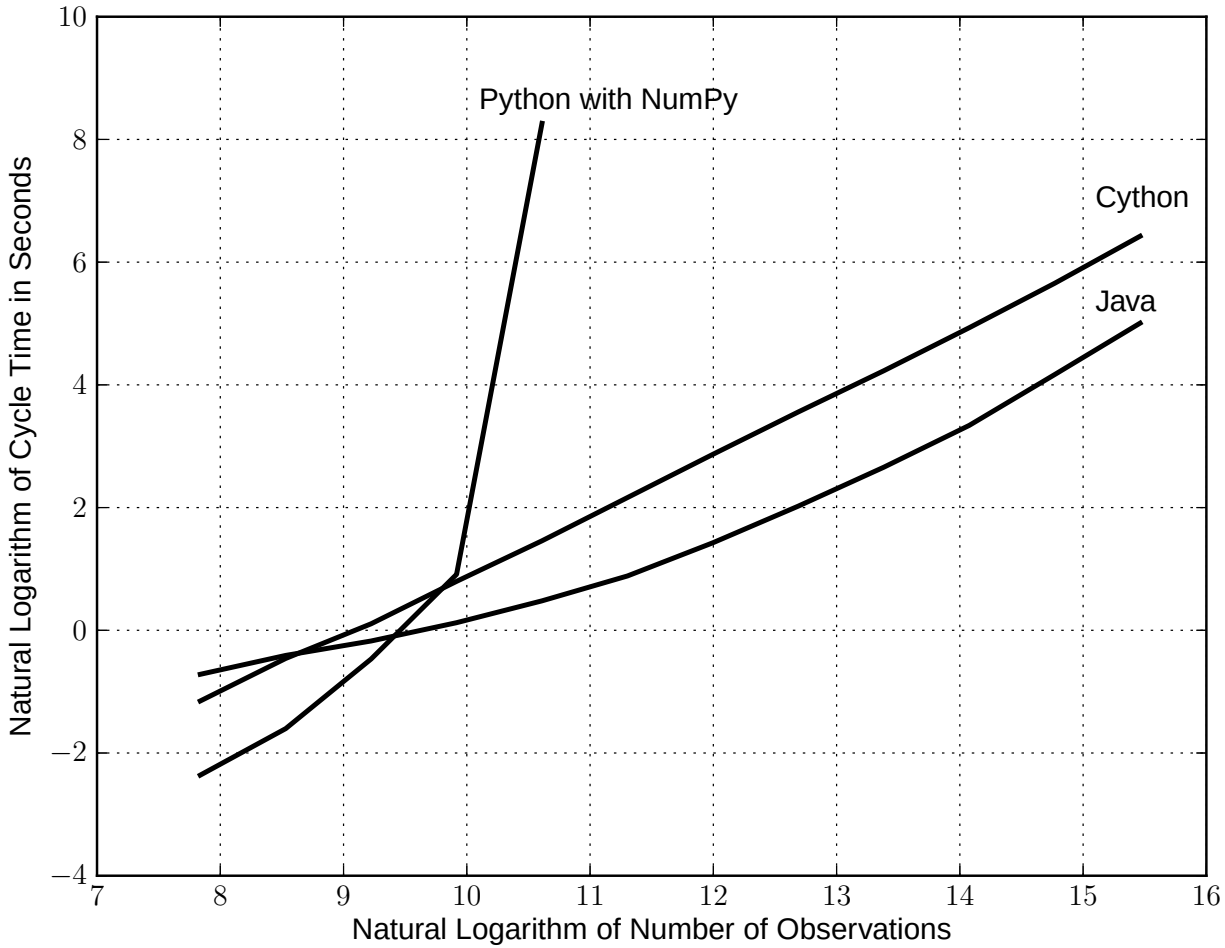
Estimated and True Data-Generating Processes



Scalability

- Does this method scale?
- I created duplicates of the previous process by doubling each data input file; that is, the data sets have $N = 2534, 2 \times 2534; 2^2 \times 2534, \ldots, 2^{16} \times 2534 = 166,068,224$ observations.
- In words, the maximum of the total number of events is $65,536,000$.
- Although straight Python with NumPy struggles to process $16,000$ events, both Cython and Java can deal effectively with over 65 million events—taking less than ten minutes to train the model.

Comparison of Cython, Java, and Python



Numerical Methods: Newton-Raphson

- Take a Taylor-series expansion of the gradient vector of the loss function evaluated at some initial point $\hat{\boldsymbol{\theta}}_0$ and set that to the zero vector:

$$\mathbf{0}_M = \mathbf{g}(\boldsymbol{\theta}) = \mathbf{g}(\hat{\boldsymbol{\theta}}_0) + \nabla_{\boldsymbol{\theta}}\mathbf{g}(\hat{\boldsymbol{\theta}}_0)(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_0) + \mathbf{R}_2.$$

- Solving yields the following recursion:

$$\hat{\boldsymbol{\theta}}_1 = \hat{\boldsymbol{\theta}}_0 - [\nabla_{\boldsymbol{\theta}}\mathbf{g}(\hat{\boldsymbol{\theta}}_0)]^{-1}\mathbf{g}(\hat{\boldsymbol{\theta}}_0).$$

Complexity Issues

- To implement Newton's method requires inverting the matrix of second partial derivatives $\nabla_{\boldsymbol{\theta}} \mathbf{g}(\hat{\boldsymbol{\theta}}_0)$, which is often referred to as the *Hessian matrix*.
- In the logit case, the Hessian matrix has a convenient closed-form expression, namely,

$$\begin{aligned} \nabla_{\boldsymbol{\theta}\boldsymbol{\theta}} L(\boldsymbol{y}, \mathbf{X}, \boldsymbol{\theta}) &= \left[\frac{1}{N} \sum_{n=1}^{N} \frac{\partial^2 \ell(y_n, \mathbf{x}_n \boldsymbol{\theta})}{\partial \theta_j \partial \theta_k} \right] \\ &= \frac{1}{N} \sum_{n=1}^{N} h(\mathbf{x}_n \boldsymbol{\theta})[1 - h(\mathbf{x}_n \boldsymbol{\theta})] \mathbf{x}_n \mathbf{x}_n^\top. \end{aligned}$$

- Inverting the Hessian matrix is an $\mathcal{O}(M^3)$ calculation, which can be computationally arduous when M is large.

Numerical Methods: Conjugate Gradient

- The *conjugate-gradient* method offers a suitable alternative in convex problems like this one.
- Introduce the direction vector

$$\mathbf{d}(\boldsymbol{\theta}) = -\frac{\mathbf{g}(\boldsymbol{\theta})}{\|\mathbf{g}(\boldsymbol{\theta})\|_2}.$$

- For notational parsimony denote the direction and gradient vectors by $\mathbf{d}_r$ $\mathbf{g}_r$ when evaluated at $\hat{\boldsymbol{\theta}}_r$.

Conjugate Gradient Recursions

- For $r = 0, 1, 2, \ldots$ and for a positive definite matrix $\mathbf{Q}$, while $\mathbf{g}_r \neq \mathbf{0}_M$, calculate the following sequence:

$$\alpha_r = \frac{\mathbf{g}_r^\top \mathbf{g}_r}{\mathbf{d}_r^\top \mathbf{Q} \mathbf{d}_r}$$
$$\hat{\boldsymbol{\theta}}_{r+1} = \hat{\boldsymbol{\theta}}_r + \alpha_r \mathbf{d}_r$$
$$\mathbf{g}_{r+1} = \mathbf{g}_r - \alpha_r \mathbf{d}_r$$
$$\omega_r = \frac{\mathbf{g}_{r+1}^\top \mathbf{g}_{r+1}}{\mathbf{g}_r^\top \mathbf{g}_r}$$
$$\mathbf{d}_{r+1} = \mathbf{g}_{r+1} + \omega_r \mathbf{d}_r.$$

Operational Problems

- Suppose not every value in the set $\{1, 2, \ldots, J\}$ is observed.
- Numerical methods fail because the Hessian, or it approximation, is not of full rank.
- Two potential solutions:
  1. pad the data set with fake observations;
  2. add a diagonal matrix whose elements are positive values.