Customer Lifetime Value modeling with JAX

yukoga@

Photo by Susan Jane Golding

Problem statement / Motivation

Customer Lifetime Value (CLV / CLTV) ...

A metric that businesses see how much profit their customers bring to them over time.

Challenges in existing solutions

- Some solutions using "pesuedo" CLTV (e.g. churn probability within a specific time frame). i.e. Would like to know CLTV for "**any**" time/day.
- Too late to take an action to customers who is likely to churn.
- Hard to incorporate customer characteristics into the model.

CLTV modeling problem's class

purchase behavior and churn observation type	Contractual (Customer 'death' can be observed)	Non contractual (Customer 'death' is unobserved)
Continuous (Purchases can happen at any time.)	Shopping with credit card	Retail ecommerce
Discrete (Purchases occur at fixed periods or freqency.)	Subscription Insurance / Finance	Nail salons
Common methodology	Survival Analysis	BTYD model

How to model expected CLTV

For contractual model, the expectation value of CLTV can be written as follows (Fader, Peter, & Bruce (2007a)):

$$E[CLTV] = \sum_{t=0}^{\infty} \frac{m}{(1+d)^t} s(t) \quad \dots \quad (1)$$

- t : Discrete time.
- m : Monetize value.
- $s(t)$: survival function at time t.
- d : discount rate reflecting the time value of money.

How to model expected CLTV

Here's an example for the following scenario:

- We have 1,000 customers at t_0 (e.g. year 1), 670 at t_1, 480 at t_2, 350 at t_3 ...
- $m = \$50/year$
- $d = 15\%$

$$E[CLTV] = 50 + \frac{50}{1.15} \cdot \frac{670}{1000} + \frac{00}{1.15^2} \cdot \frac{480}{1000} + \frac{50}{1.15^3} \frac{350}{1000} \quad \ldots$$

- For given observed data, we can calculate cLTV using the eq. (1) as above.
- But the problem is, we don't have the right survival function $s(t)$ for new customers whose CLTV we're going to predict.

Geometric-beta model (Fader, Peter & Bruce (2007a))

Assume customer lifetime follows a geometric distribution because customers can churn only once.

- Churn probability : θ
- Retention probability for customer : $1 - \theta$
- Churn probability at time t :

$$P(T = t|\theta) = \theta(1-\theta)^{t-1} \quad \dots \quad (2)$$

Geometric-beta model (cont'd)

For given Churn and retention probability, survival rate and retention rate at time t as follows:

- Survival rate:

$$s(T=t|\theta) = (1-\theta)^t \quad \dots \quad (3)$$

We model the heterogeneity of θ as a beta distribution (Since θ is bounded between $[0, 1]$ as it's probability).

- Prior distribution for θ:

$$f(\theta|\alpha, \beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)} \quad \dots \quad (5)$$

- α, β : Given parameters for the prior distribution.

Geometric-beta model (cont'd)

If we get the true θ from inference using $P(T|\theta)$ and prior $f(\theta)$, we'll get LTV through $s(t)$:

$$P(T|\theta) f(\theta) \, (eq.2\&5) \underset{inference}{\rightarrow} \hat{\theta} \rightarrow s(T|\hat{\theta}) \, (eq.3) \rightarrow LTV \, (eq.1) \quad \ldots \quad (4)$$

Maximum Likelihood Estimation with JAX

A naive way to get inferenced θ is Maxium Likelihood Estimation a.k.a. MLE. Funadamental code with JAX for MLE is as follows:

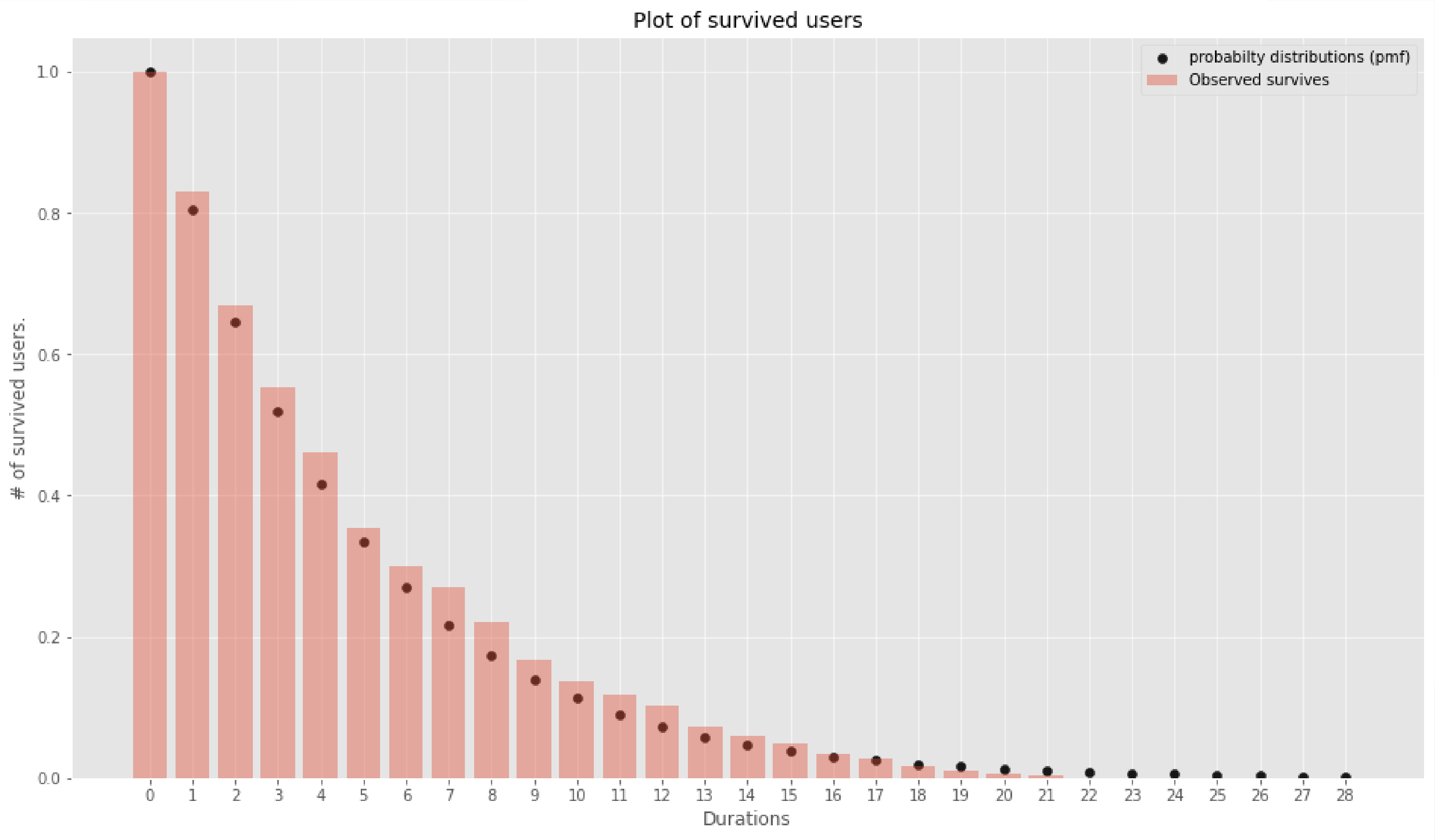
```
from jax.numpy as jnp
from jax.scipy.stats import geom

def loglikelihood(y, w):
    return jnp.sum(geom.logpmf(y, w))

def loss(y, w):
    size = y.shape[0]
    return (-1.0 * loglikelihood(y, w)) / size  # negative loglikelihood
```

Maximum Likelihood Estimation with JAX (cont'd)

```
from jaxopt import ScipyBoundedMinimize

m = ScipyBoundedMinimize(
    fun=loss,
    method="l-bfgs-b",
    options={...},
)
lb, ub = 0.00001, 0.99999
result = m.run(param, (lb, ub), (y, _))

## result.x --> parameter

## For continuous distributions, there's minimize function in jax.scipy.optimize
## result = minimize(loss, param, (y, _), method="BFGS", options={...})
```

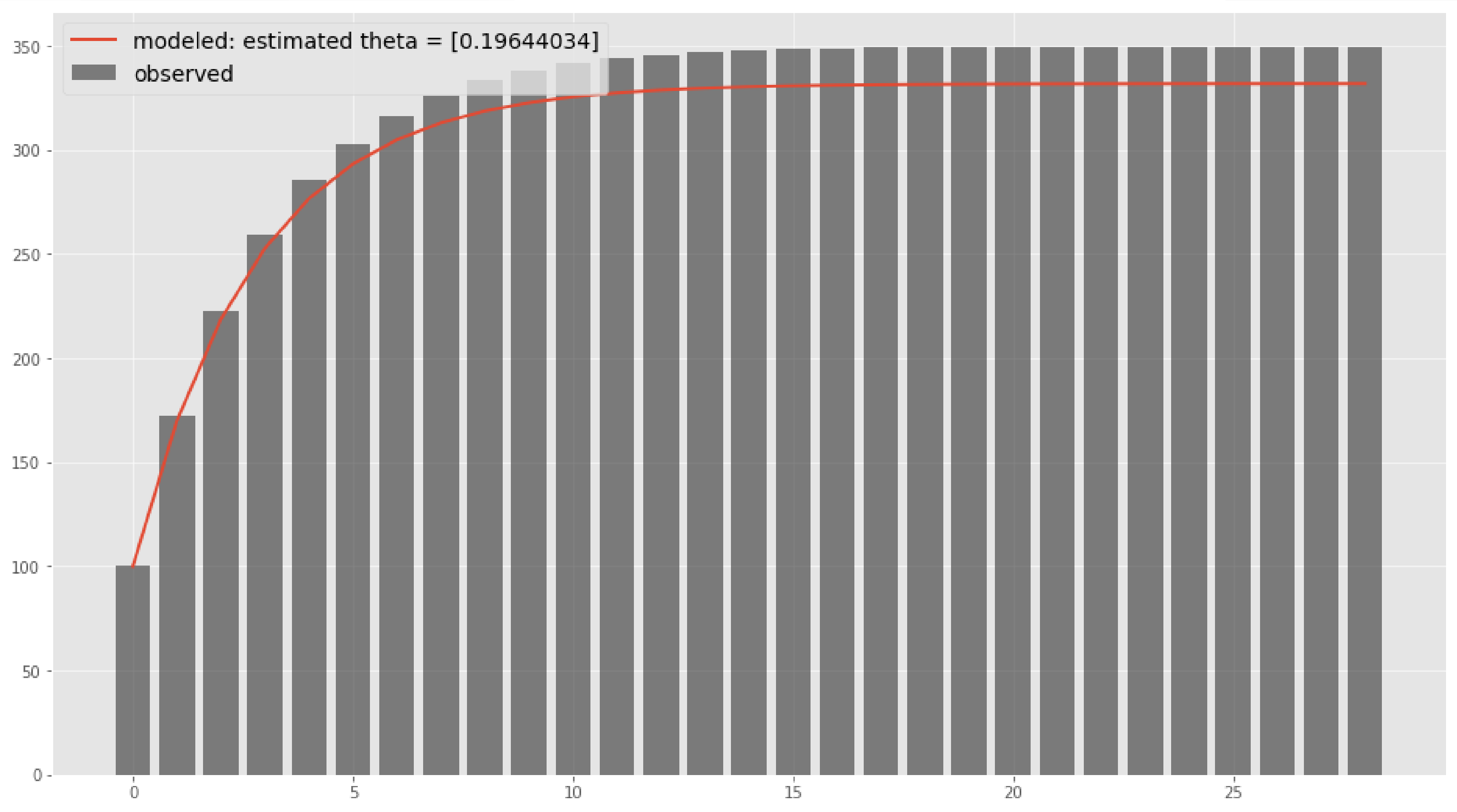
Maximum Likelihood Estimation with JAX (experiment)

Survival function observations and estimated $(\theta = 0.2)$



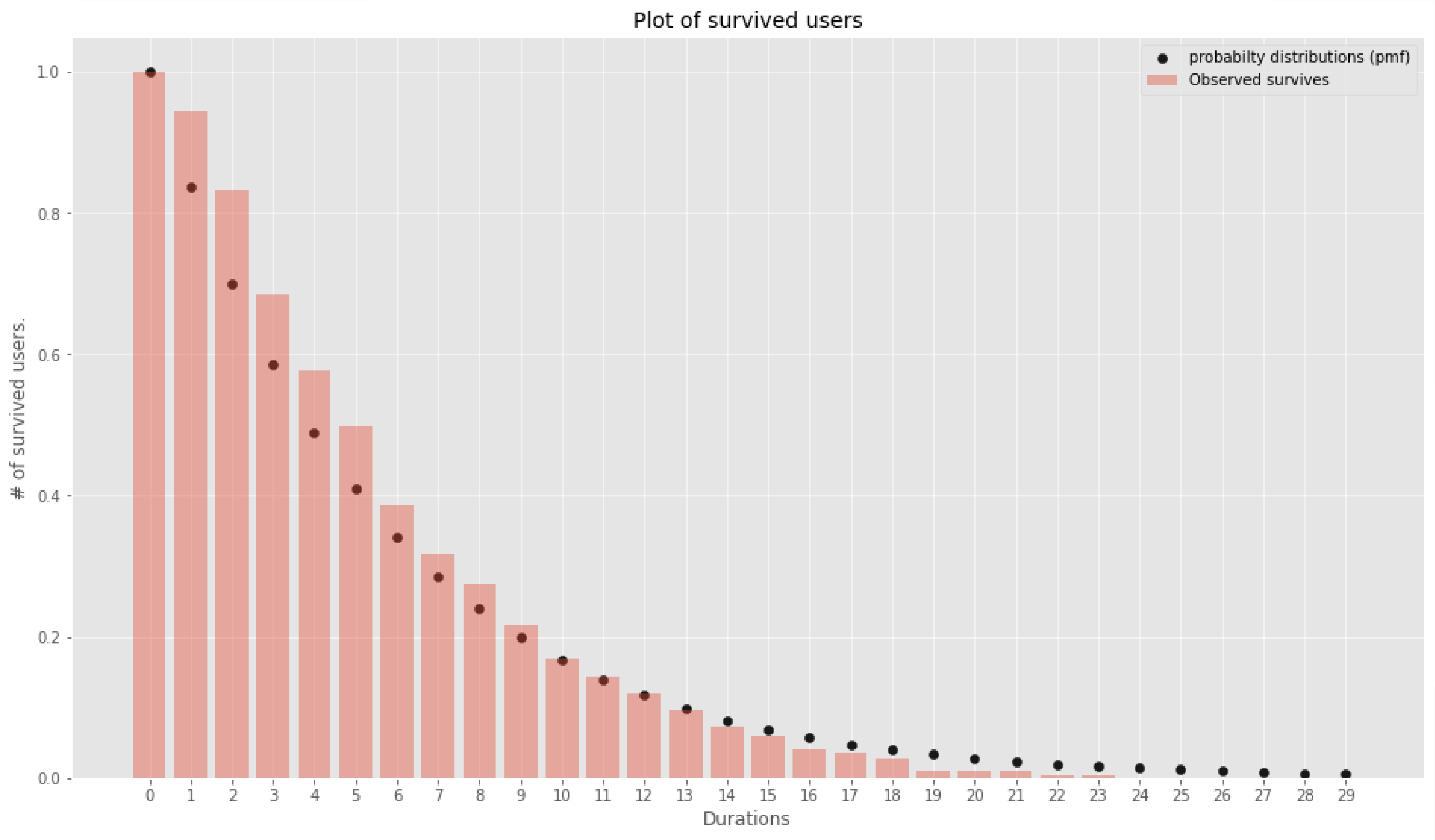
Maximum Likelihood Estimation with JAX (experiment: cont'd)

LTV observations and estimated $(\theta = 0.2)$



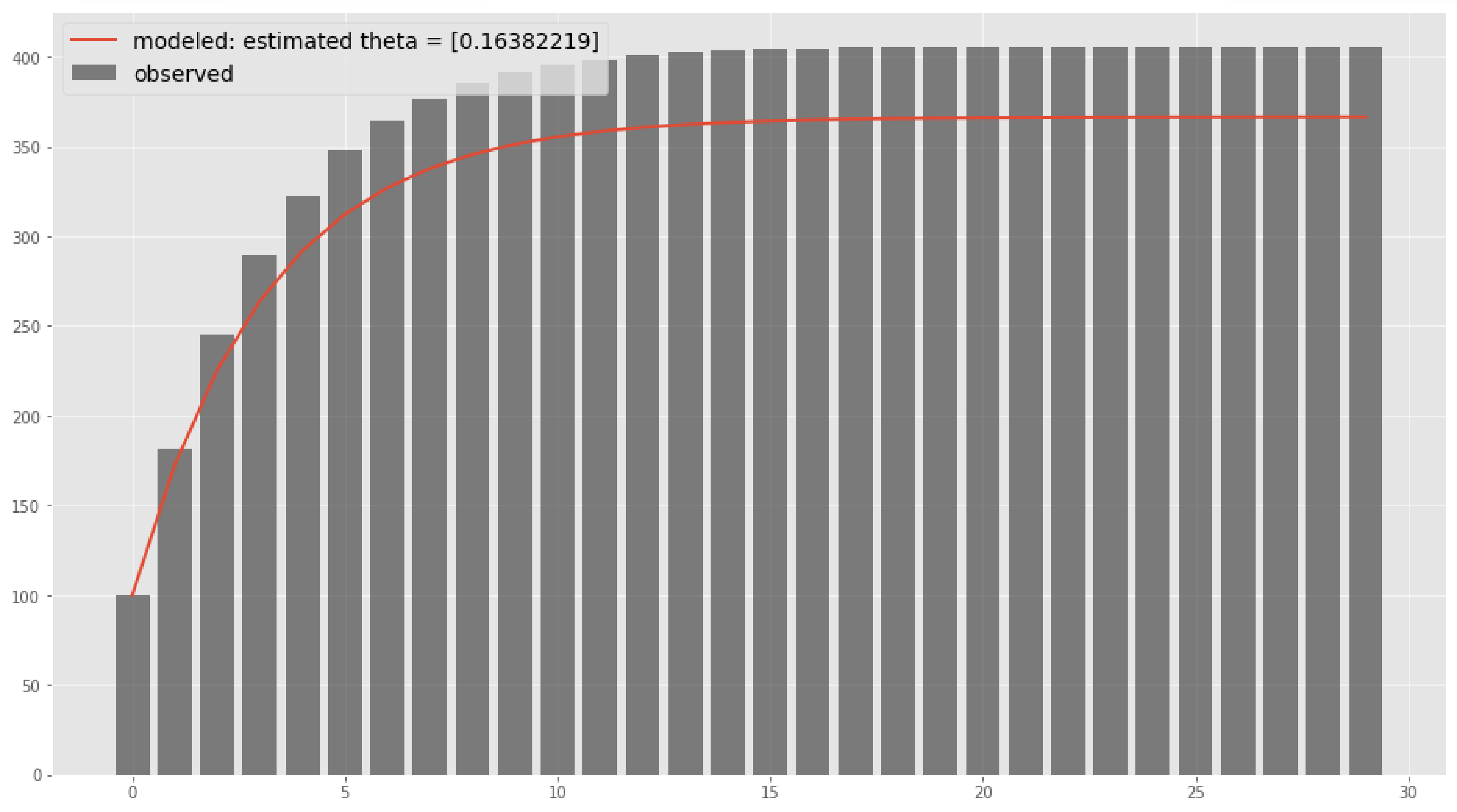
Maximum Likelihood Estimation with JAX (experiment: cont'd)

Survival function observations and estimated $(Geom(\theta = 0.2) + Poi(\lambda = 1.0))$



Maximum Likelihood Estimation with JAX (experiment: cont'd)

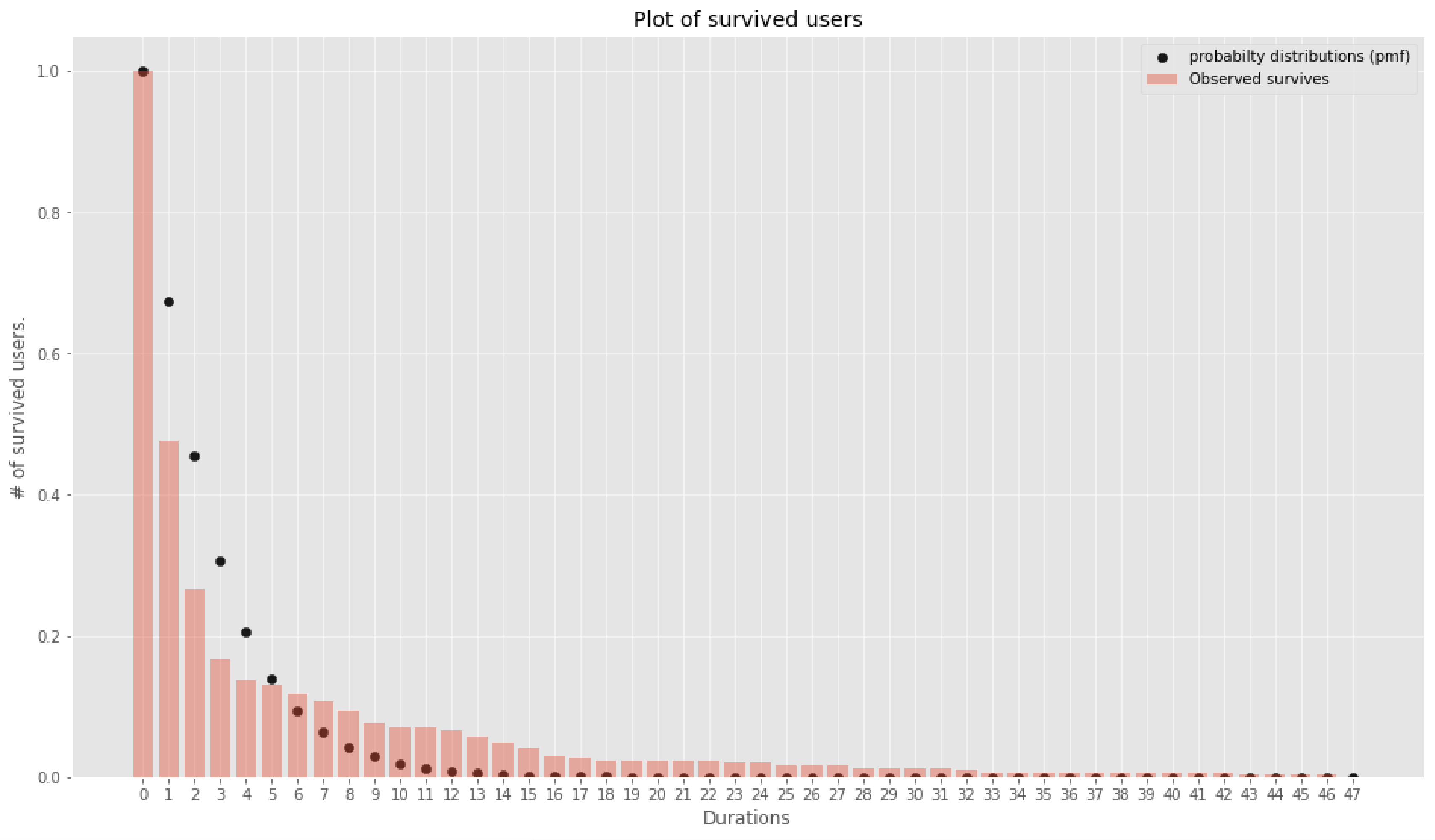
LTV observations and estimated $(Geom(\theta = 0.2) + Poi(\lambda = 1.0))$



Maximum Likelihood Estimation with JAX (experiment: cont'd)

Survival function observations and estimated for Mixed distributions

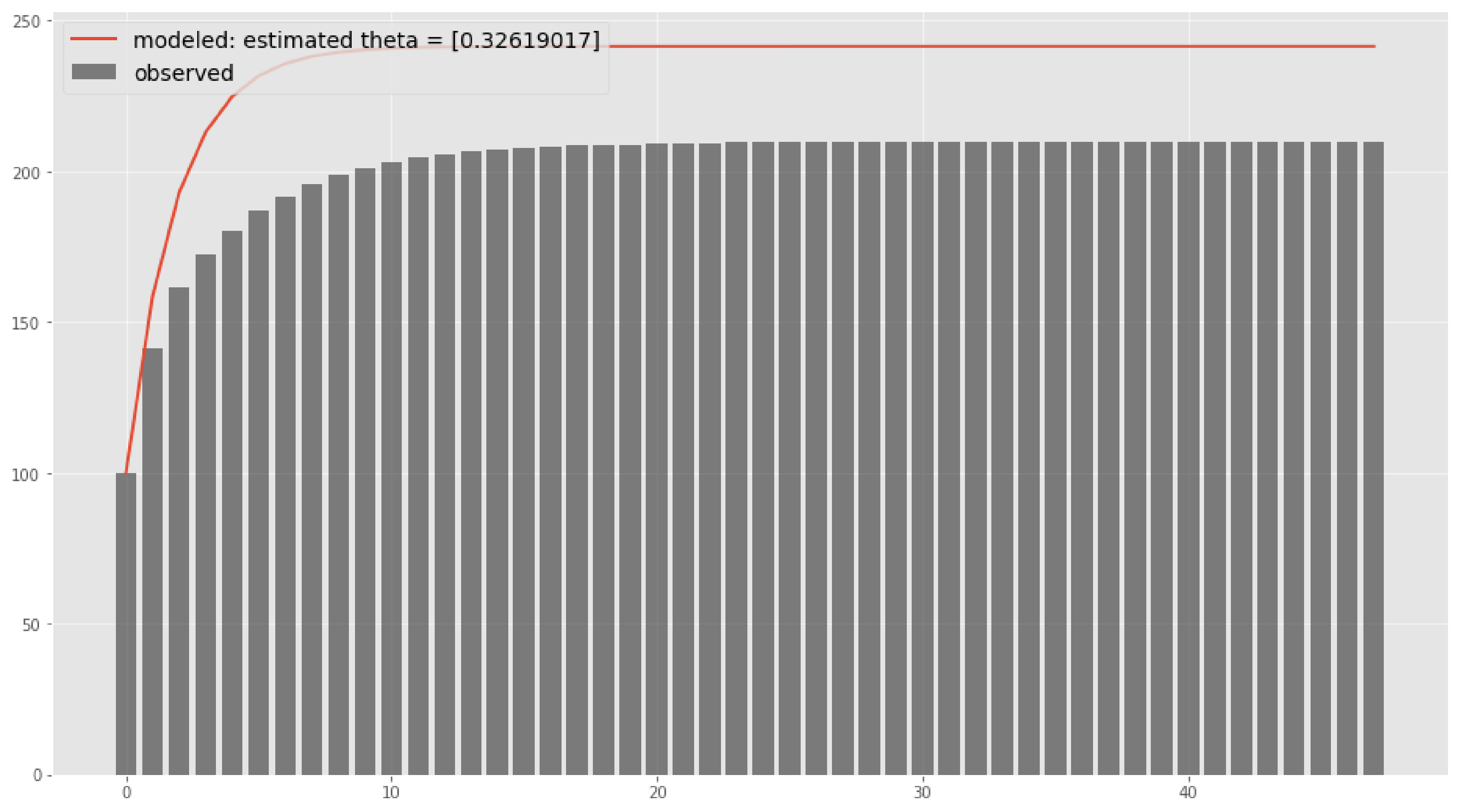
$$(Geom(\theta = 0.7) + Geom(\theta = 0.1))$$



Maximum Likelihood Estimation with JAX (experiment: cont'd)

LTV observations and estimated for Mixed distributions

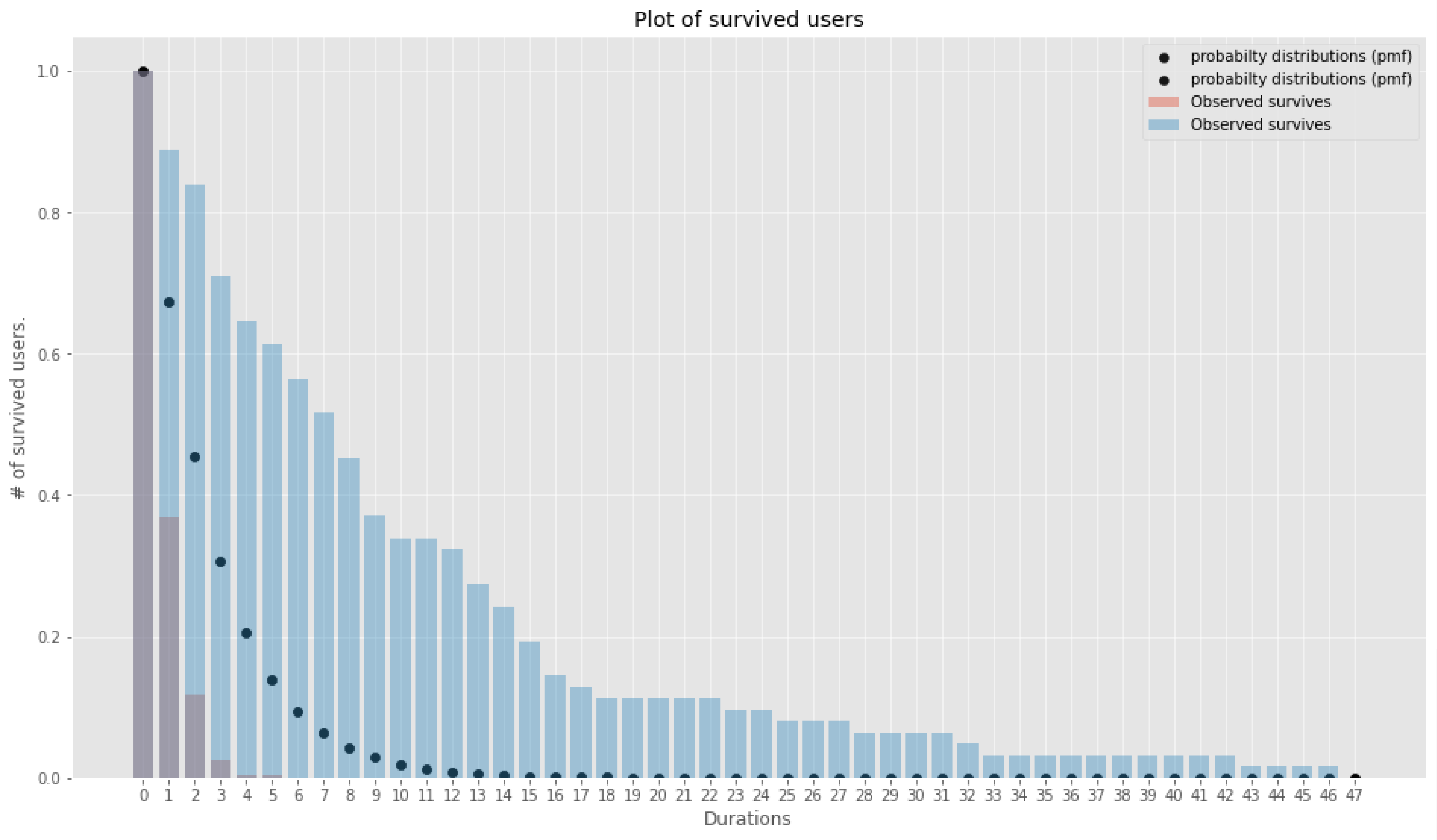
$$(Geom(\theta = 0.7) + Geom(\theta = 0.1))$$



Maximum Likelihood Estimation with JAX (experiment: cont'd)

Survival observations and estimated for Mixed distributions

$$(Geom(\theta = 0.7) + Geom(\theta = 0.1))$$



Bayesian inference with JAX

We need to estimate the heterogeneity of θ based on given data accordingly. Bayesian inference will work for that case. Assume a beta distribution as a prior of θ (Since θ is bounded between $[0, 1]$ as it's probability).

- Prior distribution for θ_i :

$$f(\theta_i|\alpha, \beta) = \frac{\theta_i^{\alpha-1}(1-\theta_i)^{\beta-1}}{B(\alpha, \beta)} \quad \dots \quad (5)$$

- α, β : Latent parameters contains customer's characteristics as follows:

Bayesian inference with JAX (experiment: cont'd)

Training data for Bayesian analysis as follows:

	Segment (= feature)	Churn date (= target)
1	A	1
2	B	7
3	A	3
4	B	5
..	...	...

Bayesian inference with JAX (experiment: cont'd)

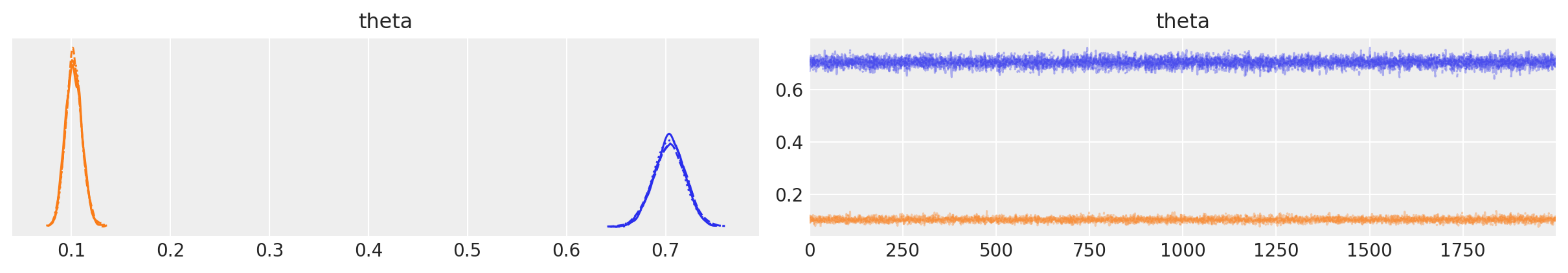
Here is PyMC4 code to get inferenced θ. Funadamental code with PyMC4 for baysian inference is as follows:

```
import pymc as pm

model = pm.Model()

with model:
    x_ = pm.Data('features', X_train.values, mutable=True)
    theta = pm.Beta('theta', 1., 1., shape=[X_train.max()+1])
    obs = pm.Geometric('obs', theta[x_], observed=y_train.values)
    idata = pm.sample(draws=2000, target_accept=0.9)
```

Bayesian inference with JAX (experiment: cont'd)

Here is PyMC4 code to get inferenced θ. Funadamental code with PyMC4 for baysian inference is as follows:

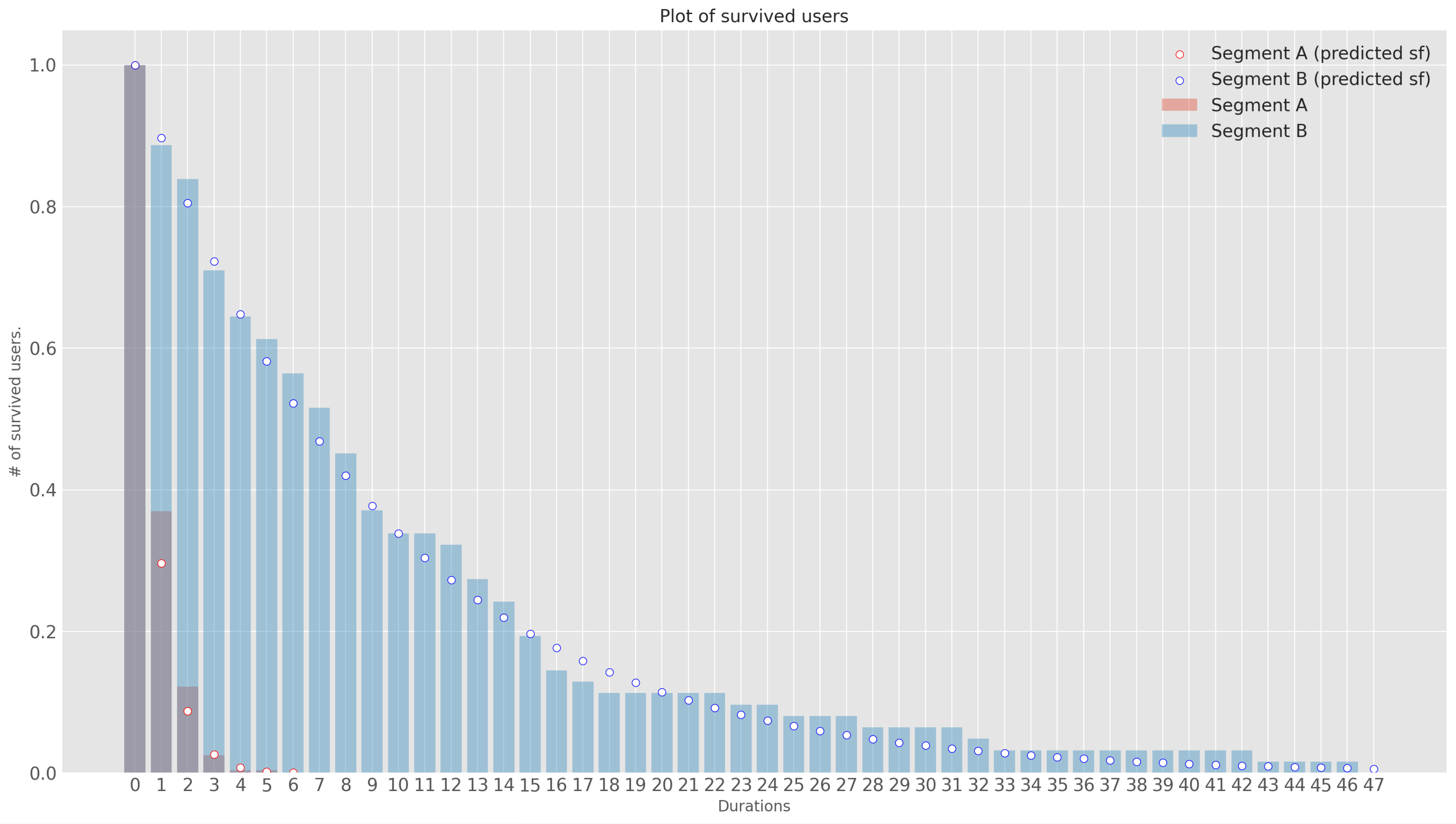
```
import arviz as az

az.plot_trace(idata, ['theta'])
```



Bayesian inference with JAX (experiment: cont'd)

Survival observations and estimated for Mixed distributions

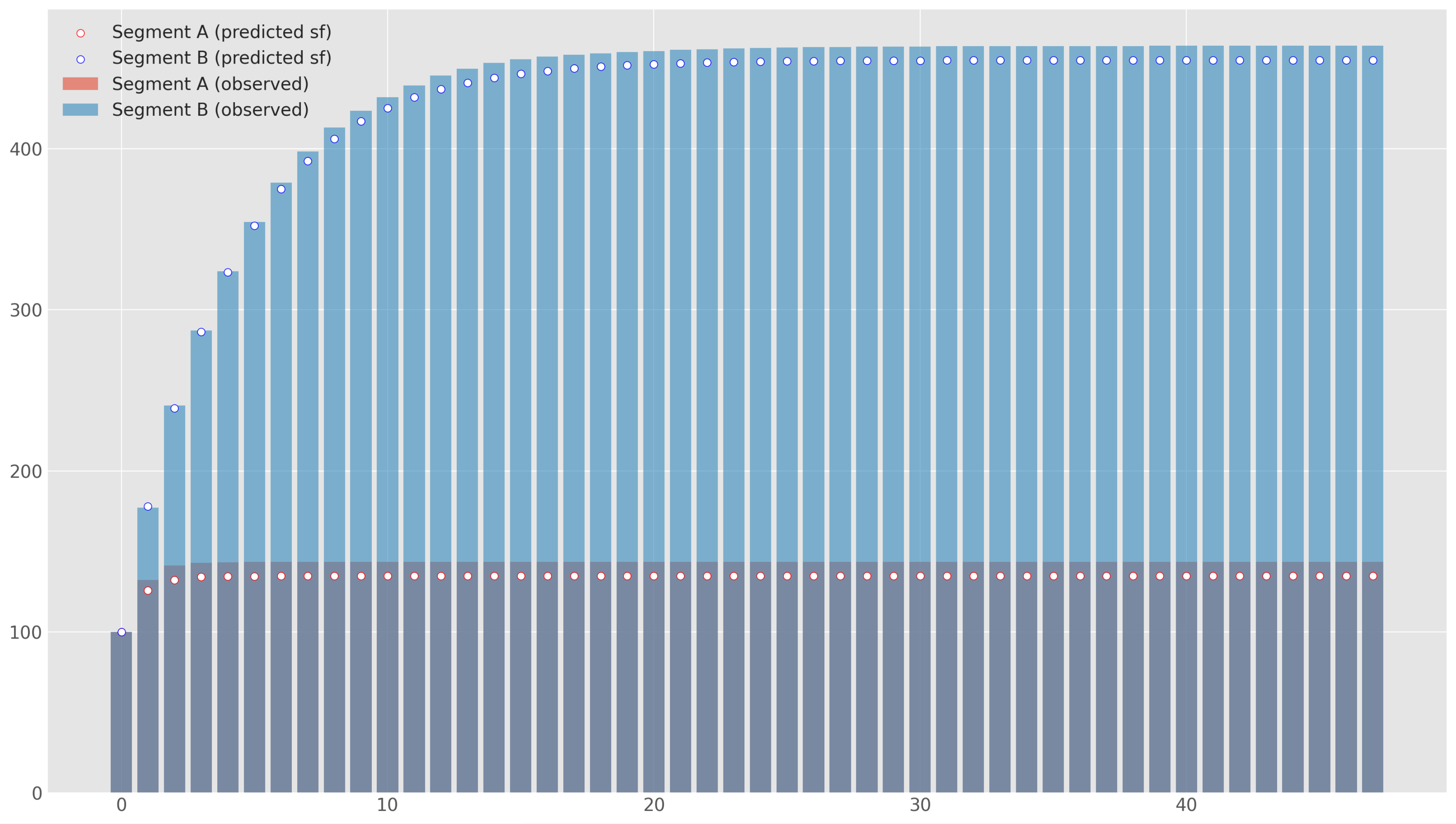
$$(Geom(\theta = 0.7) + Geom(\theta = 0.1))$$



Bayesian inference with JAX (experiment: cont'd)

LTV observations and estimated for Mixed distributions

$$(Geom(\theta = 0.7) + Geom(\theta = 0.1))$$




Thanks.
Photo by Tobi Gaulke