GENESYS

Counting Your Customers the Hard Way

Modeling Software-as-a-Service Usage and Churn with PyMC3

-- Anthony Alford

Intro

- https://github.com/anthonyalford/churn-and-usage
- Please interrupt and ask questions!
- Software Engineer at Genesys
- Genesys is the leader in cloud and on-premises contact center solutions

Understanding Customer Behavior with Data

- Genesys is a customer experience company
- Want to help our customers understand *their* customer behavior
- Started with our *own* customer behavior: predicting churn
- Focus: Enterprise Software-as-a-Service customers

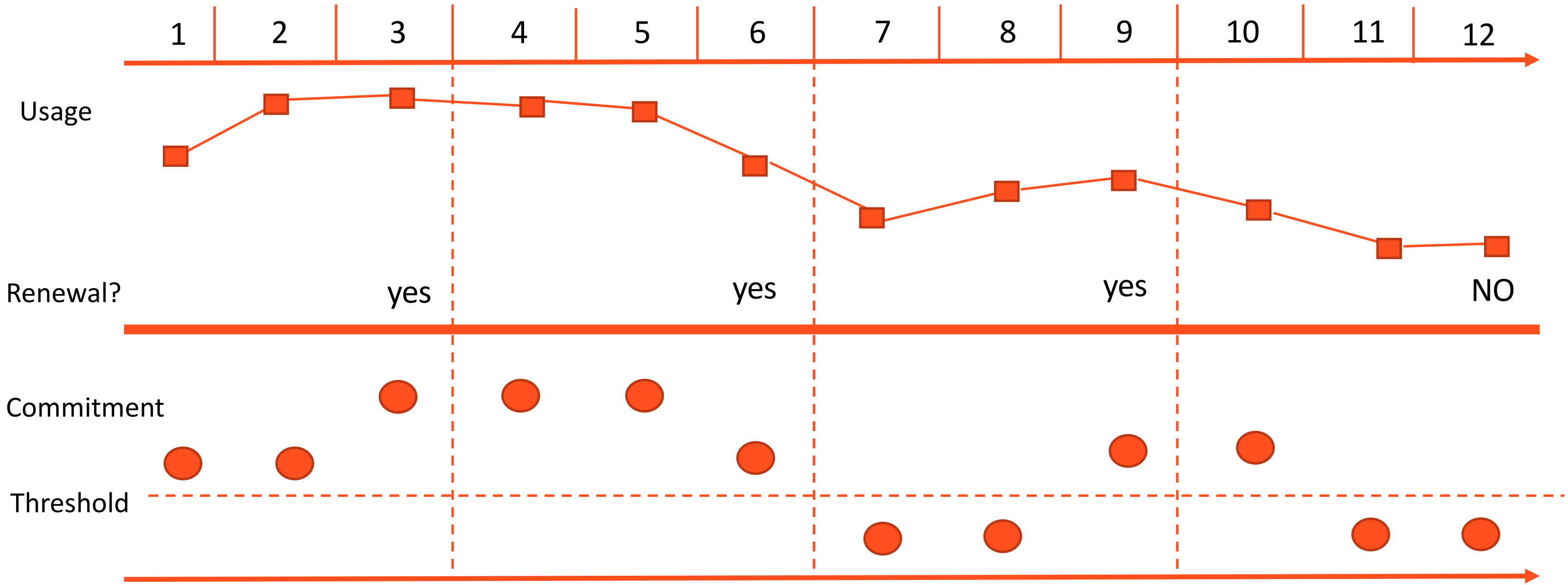
Churn and Customer Lifetime Value (CLV)

- Churn is when a customer leaves you
  - With contractual customers, predicting churn is predicting non-renewal
- Customer Lifetime: how long the customer was with you
  - With contractual customers, this is a multiple of contract period
- Customer Lifetime Value: net profit from that customer over lifetime
  - Ex: (recurring revenue – recurring costs) * lifetime
- With SaaS, cost is a function of *usage*
  - Revenue might be also (pay for usage model)
- So: knowing both churn and usage are key to predicting CLV

Research

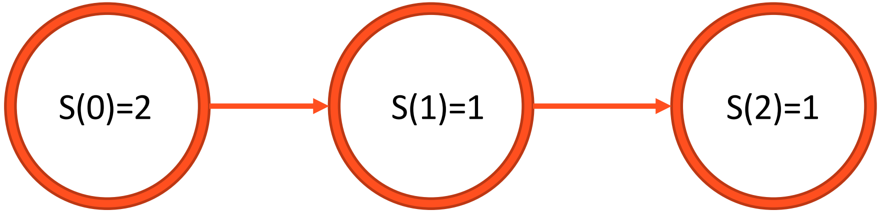
- Great talk on CLV by Van Dyke and Gauthier at PyData Seattle 2017:
  - “Implementing and Training Predictive Customer Lifetime Value Models in Python”
- Reference: “Counting Your Customers the Easy Way” by Fader, Hardie, & Lee
- Ascarza and Hardie: “Modeling Churn and Usage Behavior in Contractual Settings”
  - *“both usage and churn are functions of a latent process representing a customer's ‘commitment level.’”*
- This sounded promising: forecasting both churn and usage
  - And “commitment” might identify *loyal* customers as well as dissatisfied ones
- Let’s do it!

Usage, Renewal, and Commitment



Commitment: Markov Process

- Initial state drawn from Categorical distribution, with initial probability vector
- Next state drawn from Categorical, with transition probability matrix

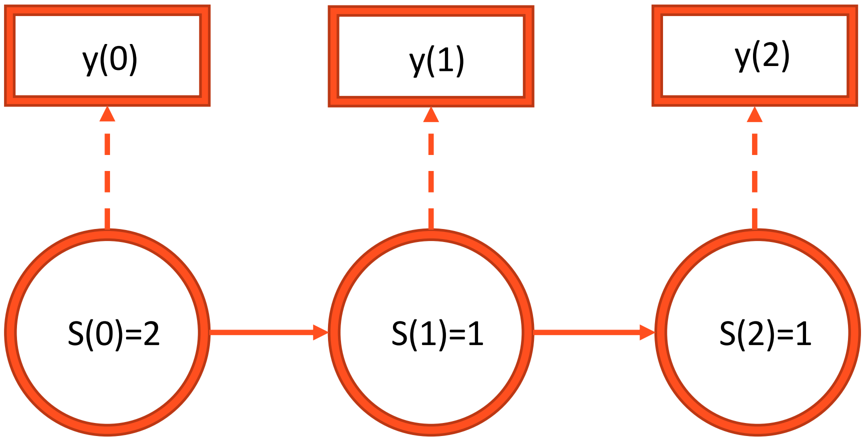


Transition Matrix

current state / next state probability

$$
\begin{array}{c}
\\ 0 \\ 1 \\ 2
\end{array}
\begin{array}{c}
\begin{array}{ccc} 0 & 1 & 2 \end{array} \\
\begin{pmatrix} 0.85 & 0.1 & 0.05 \\ 0.2 & 0.55 & 0.25 \\ 0.15 & 0.4 & 0.45 \end{pmatrix}
\end{array}
$$

Usage: Poisson Process



- Observation for customer *i* at time *t* drawn from Poisson distribution with λ_{it} composed of
  - State-specific factor
  - Customer-specific factor
- Each state *k* has a global usage factor θ_k
- Each customer *i* has usage factor α_i
- $\lambda_{it} = \alpha_i * \theta_k$

Notebook

Using the model to generate data

"Some people, when confronted with a problem, think 'I know, I'll use regular expressions.' Now they have two* problems."

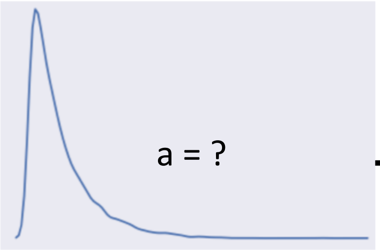
-- Jamie Zawinski

“s/regular expressions/probabilistic programming/g”

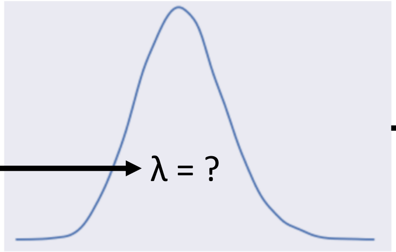
-- Me

* point estimate

Something Simple



Exponential (prior)

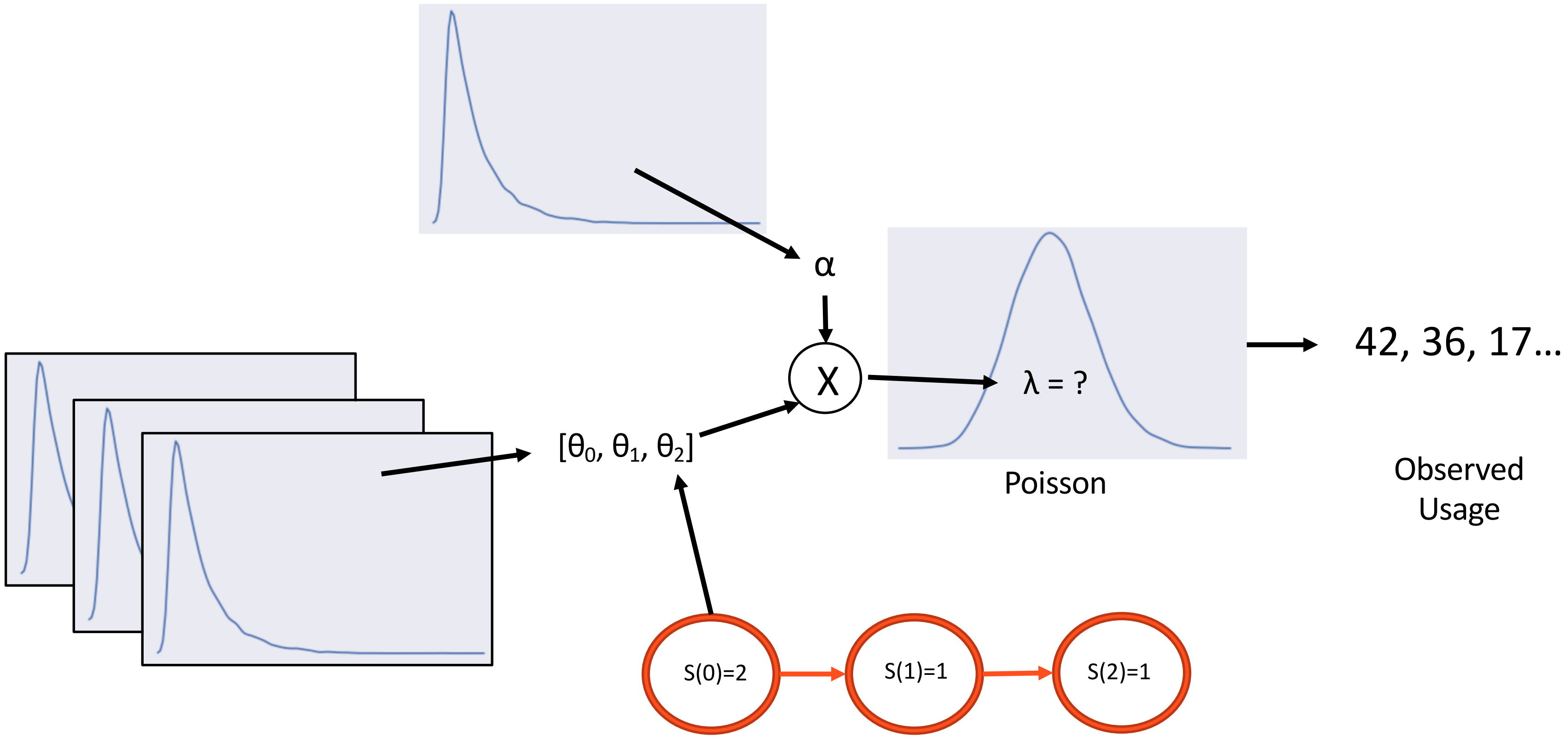


Poisson

42, 36, 17

Observed
Usage

Not So Simple

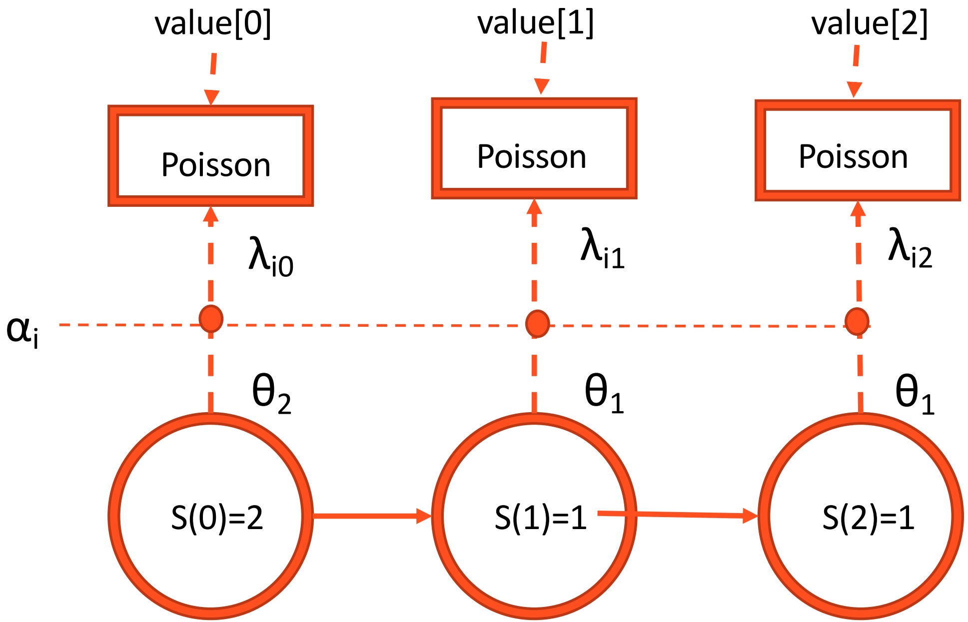


Notebook

Custom Distributions for PyMC3

GENESYS™

Custom Distribution: Usage



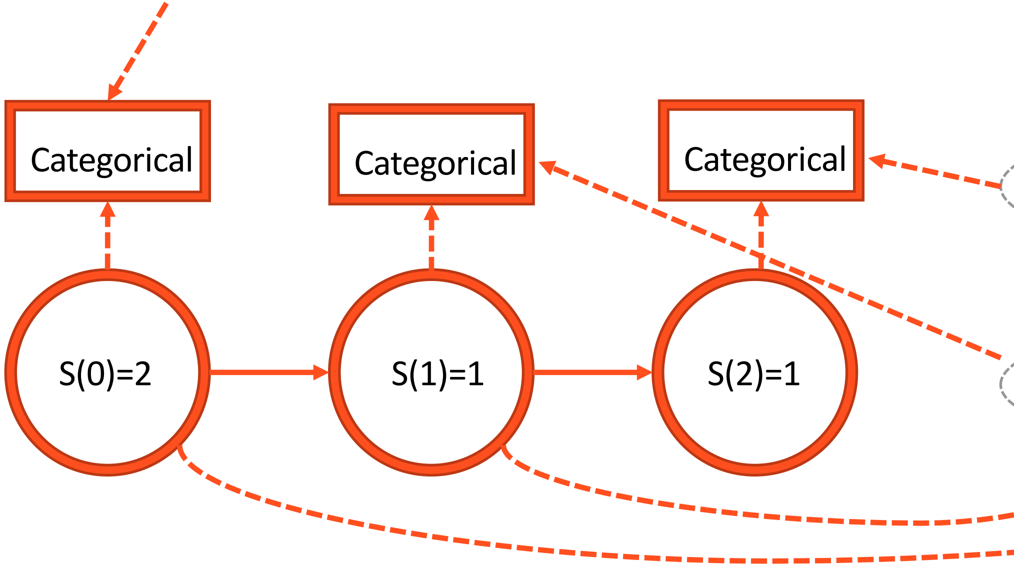
- For customer *i*, we need to calculate λ for each time period
- In time period *t*, customer is in state *k*, so lookup θ_k
- $\lambda_{it} = \alpha_i * \theta_k$
- Create tensor of λ and values, hand that off to built-in Poisson distribution
- Sum it up

Custom Distribution: Commitment

$$
\begin{array}{cccc}
 & 0 & 1 & 2 \\
Q= & \multicolumn{3}{c}{\begin{bmatrix} 0.85 & 0.1 & 0.05 \end{bmatrix}}
\end{array}
$$

Pi

$$
\begin{array}{c|c}
\text{current state} & \text{next state probability} \\
 & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 0.85 & 0.1 & 0.05 \\ 0.2 & 0.55 & 0.25 \\ 0.15 & 0.4 & 0.45 \end{bmatrix}
\end{array}
$$



Notebook

Forecasting and insight

GENESYS™

Notebook

Using real data and evaluating models

Caveats

- Didn’t work with more than ~50 customers!
- Didn’t work on AWS Deep Learning Ubuntu (with real customer data)
- Implementation expects “rectangular” data (same # months per customer)
- Model somewhat simplistic (authors address in another paper) :
  - Same commitment dynamics per customer
  - No covariates
  - No seasonality

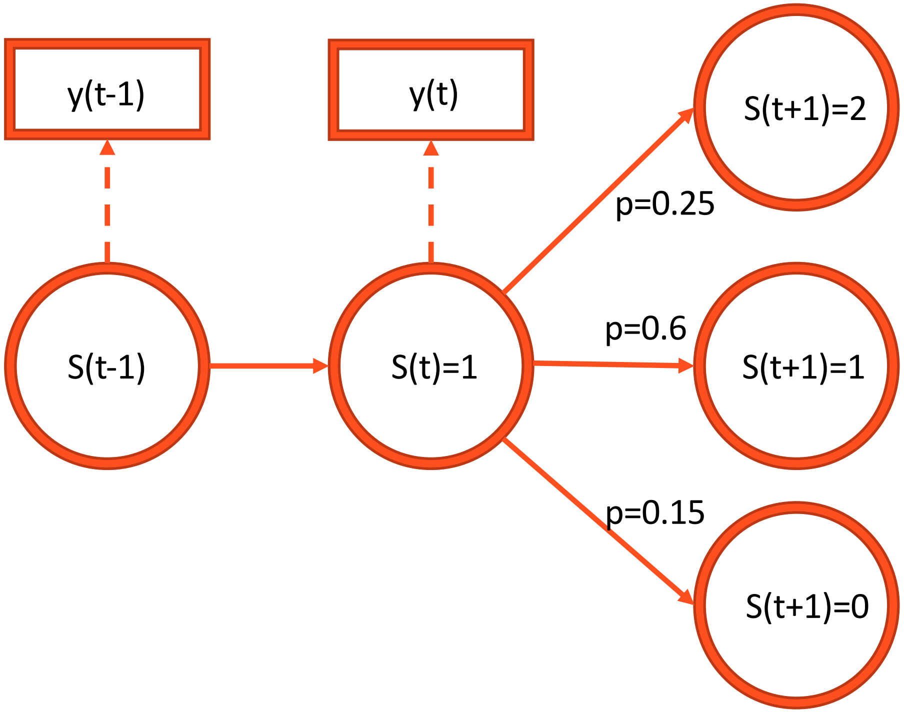
Thank You

https://github.com/anthonyalford/churn-and-usage

From Van Dyke and Gauthier

	Non-Contractual	Contractual
Continuous Purchases	• movie rentals • grocery purchases • Amazon.com	• Costco memberships • credit cards
Discrete Purchases	• prescription refills • charity fund drives • event attendance	• fitness clubs • streaming services (Netflix) • most cell phone plans

The Model



- Next state drawn from Categorical distribution, with global transition probability matrix
- Observation at time t drawn from Poisson distribution with lambda composed of
  - State-specific factor
  - Customer-specific factor