Probabilistic Changepoint Modeling

An example report for Pittsburgh useR Group

Mikhail Popov

12 January 2019

Summary Our imaginary store receives some number of customers per day. We start an advertising campaign which takes some time to have a full effect on the rate of customers we receive daily, and then has a long-term effect after we have stopped it. Using probabilistic programming languages (PPLs), we can specify a Bayesian model and infer the hidden rates.

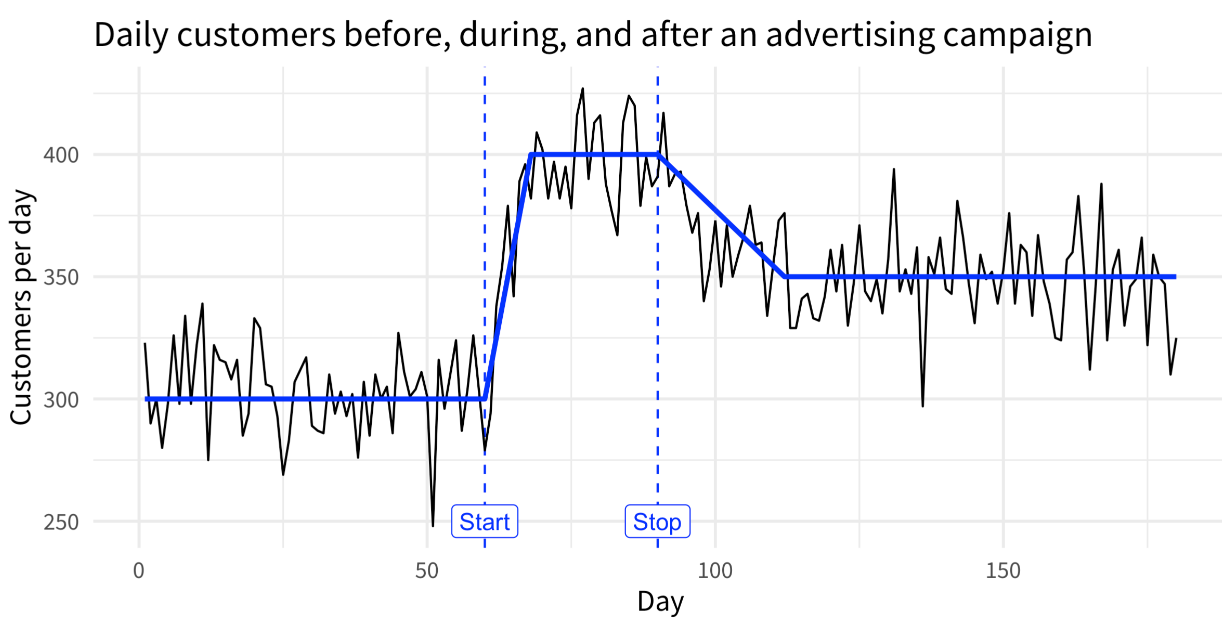
Data simulation

Suppose we are operating a store which receives a random number of customers per day, and specifically it's random according to the Poisson distribution with rate λ_1. Then we start an advertisement campaign – which takes a few days to come to full effect – that changes the rate to λ_2. When we stop the ad, it gradually loses effect but the campaign has had a long-lasting effect (e.g. we've gained new customers who come back regularly), which means our store now receives customers at rate λ_3.

Table 1: Simulation parameters

parameter	value
Rates (unknown)	
λ_1 (old normal)	300
λ_2 (temporary)	400
λ_3 (new normal)	350
Other parameters	
N (total days)	180
T_1 (ad start)	60
T_2 (ad stop)	90
d_1 (time for full effect)	7
d_2 (time to new normal)	21

In the simulation, we have two weights – w_1 and w_2 – which sum to 1 and produce gradual (albeit not smooth) transitions between the different rates. The change is a slope in this toy scenario, but it can also be a more interesting transition like a sigmoid function such as the Gompertz curve.



Modelling

We'll take a look at three similar models – $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ – which model the daily counts of customers y_t as a Poisson-distributed random variable with time-varying rate $\lambda(t)$, up to a maximum of $t = N$ days of data: $y_t \sim \text{Poisson}(\lambda(t)),\ t = 1, \ldots, N.$

The ad campaign starts on T_1 and stops on T_2. We are interested in inferring λ_1 (the rate before T_1), λ_2 (the rate between T_1 and T_2), and λ_3 (the rate after T_2). We specify $\lambda \sim \mathcal{N}(300, 100)$ and let Stan implicitly assign a default prior to β_1 and β_2.

Model 1

In the simpler model $\mathcal{M}_1$, we ignore the obvious transition periods and model the switch between the rates as immediate:

$$\lambda(t) = \begin{cases} \lambda_1, & \text{if } t \leq T_1, \\ \lambda_2, & \text{if } T_1 < t \leq T_2, \\ \lambda_3, & \text{if } t > T_2. \end{cases}$$

Model 2

In the slightly more complex model $\mathcal{M}_2$, we include the two gradual changes as slopes over d_1 and d_2 days after T_1 and T_2, respectively. We formalize this as follows:

$$\lambda(t) = \begin{cases} \lambda_1, & \text{if } t \leq T_1, \\ \lambda_2, & \text{if } T_1 + d_1 \leq t \leq T_2, \\ \lambda_3, & \text{if } t \geq T_2 + d_2, \\ \lambda_1 + \beta_1(t - T_1), & \text{if } T_1 < t \leq T_1 + d_1, \\ \lambda_2 + \beta_2(t - T_2), & \text{if } T_2 < t \leq T_2 + d_2. \end{cases}$$

Note: in this toy scenario we know d_1 and d_2, but we can also make them hidden parameters to infer from the data. We could formalize our intuition about their values by assigning them priors $\mathcal{N}(7, 2)$ and $\mathcal{N}(14, 4)$, respectively.

Model comparison

Table 2: Posterior probabilities of the three models given data and the estimated rates, calculated using the bridgesampling package.

		Estimate (95% Credible Interval)		
Model	$\Pr(\mathcal{M}\|\mathcal{D})$	λ_1 (300)	λ_2 (400)	λ_3 (350)
$\mathcal{M}_1$	0.0000	302.9 (298.6–307.1)	387.5 (380.6–394.4)	352.8 (348.9–356.9)
$\mathcal{M}_2$	1.0000	302.8 (298.5–307.1)	395.0 (387.8–402.7)	348.1 (343.8–352.5)

The Bayes factor (BF) can be used to decide between these two competing models – $\mathcal{M}_1$ (instant changes between rates) and $\mathcal{M}_2$ (gradual changes between rates) – by quantifying how much more likely the data $\mathcal{D}$ is under $\mathcal{M}_1$ vs $\mathcal{M}_2$:

$$\mathrm{BF}_{12} = \frac{p(\mathcal{D}|\mathcal{M}_1)}{p(\mathcal{D}|\mathcal{M}_2)},$$

Using **bridgesampling** (Gronau & Singmann, 2018) allows us to calculate marginal likelihoods – $p(\mathcal{D}|\mathcal{M})$ – of Stan models really easily and therefore compute the BF (see `?bf`), which – for $\mathcal{M}_1$ compared to $\mathcal{M}_2$, for example – comes out to be 0, meaning there is no evidence for choosing model 1 over model 2.

Inference results

Table 3: Inference using $\mathcal{M}_2$ (gradual changes between rates).

Parameter	Truth	Point Estimate	95% Credible Interval
Rates			
λ_1	300	302.8	(298.4, 306.9)
λ_2	400	394.9	(388.1, 403.0)
λ_3	350	348.0	(343.9, 352.5)
Differences			
$\lambda_2 - \lambda_1$	100	92.1	(84.1, 100.9)
$\lambda_3 - \lambda_2$	-50	-46.9	(-55.9, -39.1)
$\lambda_3 - \lambda_1$	50	45.2	(38.9, 51.1)

Table 3 shows the inferred differences of rates and we can see that our imaginary advertisement had a positive, statistically significant impact on how many imaginary customers our imaginary store receives per day on average, both during the campaign and long after.

References

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., … Iannone, R. (2018). Rmarkdown: Dynamic documents for r. Retrieved from https://rmarkdown.rstudio.com

Bache, S. M. (2015). Import: An import mechanism for r. Retrieved from https://CRAN.R-project.org/package=import

Gronau, Q. F., & Singmann, H. (2018). Bridgesampling: Bridge sampling for marginal likelihoods and bayes factors. Retrieved from https://CRAN.R-project.org/package=bridgesampling

Henry, L., & Wickham, H. (2018). Purrr: Functional programming tools. Retrieved from https://CRAN.R-project.org/package=purrr

Müller, K. (2017). Here: A simpler way to find your files. Retrieved from https://CRAN.R-project.org/package=here

Popov, M. (2018). Wmfpar: Wikimedia foundation's product analytics reporting template. Retrieved from https://github.com/bearloga/wmf-product-analytics-report

R Core Team. (2018). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Stan Development Team. (2018). RStan: The R interface to Stan. Retrieved from http://mc-stan.org/

Wickham, H. (2016). Ggplot2: Elegant graphics for data analysis. Springer-Verlag New York. Retrieved from http://ggplot2.org

Wickham, H., François, R., Henry, L., & Müller, K. (2018). Dplyr: A grammar of data manipulation. Retrieved from https://CRAN.R-project.org/package=dplyr

Wickham, H., & Henry, L. (2018). Tidyr: Easily tidy data with 'spread()' and 'gather()' functions. Retrieved from https://CRAN.R-project.org/package=tidyr

Xie, Y. (2014). Knitr: A comprehensive tool for reproducible research in R. In V. Stodden, F. Leisch, & R. D. Peng (Eds.), Implementing reproducible computational research. Chapman; Hall/CRC. Retrieved from http://www.crcpress.com/product/isbn/9781466561595

Xie, Y. (2015). Dynamic documents with R and knitr (2nd ed.). Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from https://yihui.name/knitr/

Xie, Y. (2018). Knitr: A general-purpose package for dynamic report generation in r. Retrieved from https://yihui.name/knitr/

Xie, Y., Allaire, J., & Grolemund, G. (2018). R markdown: The definitive guide. Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from https://bookdown.org/yihui/rmarkdown

Zhu, H. (2018). KableExtra: Construct complex table with 'kable' and pipe syntax. Retrieved from https://CRAN.R-project.org/package=kableExtra

Zhu, H., Tsai, T., & Travison, T. (2018). Memor: A 'rmarkdown' template that can be highly customized. Retrieved from https://CRAN.R-project.org/package=memor