How Does Netflix Use Always-Valid Inference in Software Deployments?

Michael Lindon (michael.s.lindon@gmail.com)

Netflix, Experimentation Platform

Among software engineers there is a field called DevOps ("Developer Operations"). DevOps is a set of philosophies and best practises on how best to release new software updates to users. Software engineers at Netflix deploy new code continuously into production, hundreds of times throughout day, following well defined protocols. Yet, releasing new software can be a source of anxiety for the developer, nervous about the potential risk of introducing **bugs** into the codebase, or causing **performance regressions** that buildup over time. To lessen this risk, it is common to run an **A/B "Canary" test** comparing the current software to the new software. A randomized subset of users are served the new code and performance metrics are carefully **monitored in real-time**. There is a strong desire to monitor these performance metrics, so that bugs and performance regressions are acaught *quickly*, so that the code can be reverted as soon as possible. Unfortunately, it is widespread across the industry to use repeated applications of classical "fixed"-n statistical tests, sacrificing Type-I error guarantees through continuous monitoring. It is clear that software engineers really seek new methodologies that control **Type-I error under *Optional Stopping* and *continuation***. Our work provides an always-valid framework for developers to compare performance metrics across code versions in real-time.

Confidence Sequences for Multinomial Probabilities

We begin by writing down the Bayes factor. Consider a sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots$ of independent Multinomial$(1, \boldsymbol{\theta})$ random variables. Under the null model $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. Under the alternative model, $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha}_0)$. We choose $\alpha_{0,i} = k\theta_{0,i}$ for a concentration parameter $k \in \mathbb{R}^+$. Under equal prior probabilities, the posterior odds of the alternative model to the null model are

$$O_n(\boldsymbol{\theta}_0) = \frac{\text{Beta}(\boldsymbol{\alpha}_0 + \boldsymbol{S}_n)}{\text{Beta}(\boldsymbol{\alpha}_0)} \frac{1}{\boldsymbol{\theta}_0^{\boldsymbol{S}_n}}, \tag{1}$$

where $S_i^n = \sum_{j=1}^n x_{j,i}$ and $\boldsymbol{S}_n = (S_1^n, \dots, S_d^n) \in \mathbb{R}^d$, $\boldsymbol{v}^{\boldsymbol{w}} = \prod_i v_i^{w_i}$ to denote element-wise exponentiation of two vectors $\boldsymbol{v}$ and $\boldsymbol{w}$, and Beta to denote the multivariate Beta function $\text{Beta}(\boldsymbol{v}) := (\prod_i \Gamma(v_i))/\Gamma(\sum_i v_i)$. Under the null, $\mathbb{E}[O_n(\boldsymbol{\theta}_0)|\mathcal{F}_{n-1}] = O_{n-1}(\boldsymbol{\theta}_0)$, and via an application of Ville's Inequality

$$\mathbb{P}_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}(\exists n \in \mathbb{N} : O_n(\boldsymbol{\theta}_0) \geq 1/u) \leq u \tag{2}$$

for all $u \in [0, 1]$. A sequential p-value for testing $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ can be defined in the following way. Let $p_0 = 1$ and $p_n = \min(p_{n-1}, 1/O_n(\boldsymbol{\theta}_0))$, then It follows from this definition that

$$\mathbb{P}_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}(\exists n \in \mathbb{N} : p_n \leq u) \leq u, \tag{3}$$

By exploting the duality between p-values and confidence statements, the last result can be inverted to yield a confidence sequence for $\boldsymbol{\theta}$. Let $C_n(u) = \{\boldsymbol{\theta} \in \triangle^d : O_n(\boldsymbol{\theta}) < 1/u\}$ denote the set of parameter vectors that would not be rejected by the test at the u level, then

$$\mathbb{P}_{\boldsymbol{\theta}}(\boldsymbol{\theta} \in C_n(u) \text{ for all } n \in \mathbb{N}) \geq 1 - u \tag{4}$$

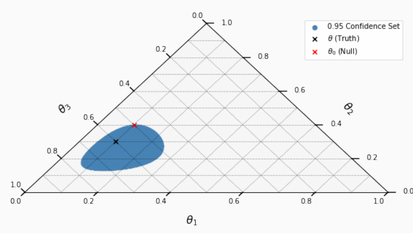
for all $u \in [0, 1]$. Confidence intervals on the individual elements of $\boldsymbol{\theta}$ can be obtained by projecting $C_n(u)$ onto the coordinate axes in the following manner. Let $j_{n,i}^+(u) = \sup\{\theta_i : \boldsymbol{\theta} \in C_n(u)\}$ and $j_{n,i}^-(u) = \inf\{\theta_i : \boldsymbol{\theta} \in C_n(u)\}$ then

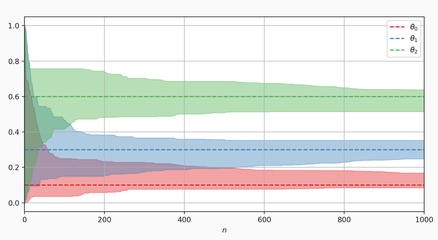
$$\mathbb{P}_{\boldsymbol{\theta}}\left(\forall i : \theta_i \in \bigcap_{n=1}^{\infty} [j_{n,i}^-(u), j_{n,i}^+(u)]\right) \geq 1 - u. \tag{5}$$

$j_{n,i}^+(u)$ and $j_{n,i}^-(u)$ can be computed by solving the following convex optimization program

$$\begin{aligned} \max \quad & \theta_i \\ \text{s.t.} \quad & c + \log u \leq \sum_i S_i^n \log \theta_i \\ & \sum_i \theta_i = 1 \end{aligned} \tag{6}$$

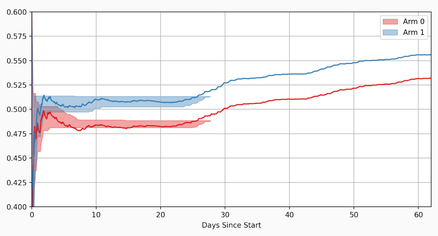
The left figure shows the confidence set $C_n(u)$ while the right figure shows confidence sequences on the individual elements of $\boldsymbol{\theta}$.

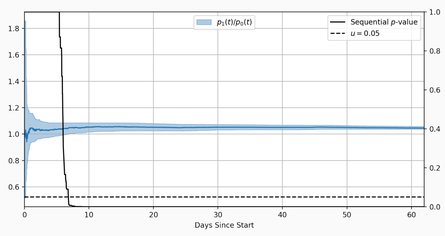




Sequential Tests of Inhomogeneous Bernoulli Processes

In payments and signup experiments, outcomes are often modelled as Bernoulli's with constant success probabilities. If these assumptions were correct, then the Binomial-Beta confidence sequences could be used to sequentially estimate the success probability of each treatment group. These assumptions are rarely true in practice, and the left figure shows what can go wrong.





The following case study is taken from a signup funnel experiment at Netflix. The left figure shows the application of the multinomial confidence sequences to estimating the conversion probabilities for arms 0 and 1. Note that the running intersection of always-valid confidence intervals becomes the empty set, and the point estimate exits the confidence sequence. This would a rare event (with probability less than α) if the stationary Bernoulli assumptions were true. This is usually a sign that the conversion probabilities are not constant, but time-varying, invalidating a commonly made assumption in conversion rate experimentation.

Suppose a new experimental unit is randomly assigned to one of d experiment arms at time t, according to assignment probabilities $\boldsymbol{\rho} \in \triangle^d$, and a Bernoulli outcome is observed. The Bernoulli probability for arm i at time t is parameterized by $p_i(t) = e^{\mu(t)}e^{\delta_i}$ so that the time-varying effect is multiplicative and common to all arms. The improvement of arm j over arm i at any time is then $p_j(t)/p_i(t) = \exp(\delta_j - \delta_i)$, and the difference on the log-scale is simply $\delta_j - \delta_i$. Suppose Bernoulli failures are ignored and arms are compared only through their counts of Bernoulli successes. The (conditional) probability that the next Bernoulli success comes from arm i is

$$\theta_i = \sigma_{\boldsymbol{\rho}}(\boldsymbol{\delta})_i := \frac{\rho_i e^{\delta_i}}{\sum_{j=1}^d \rho_j e^{\delta_j}}, \tag{7}$$

which is independent of the time-varying effect. The arm from which the next Bernoulli success arrives is, therefore, a Multinomial$(1, \boldsymbol{\theta})$ random variable, and the counts of Bernoulli successes for each arm are an ancillary statistic with respect to the time-varying nuisance parameter $\mu(t)$. Framing the problem this way allows the sequential multinomial test to perform inference on $\boldsymbol{\delta}$. Let $K_n(u) = \sigma_{\boldsymbol{\rho}}^{-1}(C_n(u))$, then

$$\mathbb{P}[\boldsymbol{\delta} \in K_n(u) \text{ for all } n \in \mathbb{N}] \geq 1 - u \tag{8}$$

Let $\mathcal{A}^d = \{\boldsymbol{a} \in \mathbb{R}^d : \sum_i a_i = 0\}$ denote the set of all d-dimensional contrasts. For all $\boldsymbol{a} \in \mathcal{A}^d$ define

$$l_{n,\boldsymbol{a}}^+(u) = \sup\{\sum_i a_i\delta_i : \boldsymbol{\delta} \in K_n(u)\},$$
$$l_{n,\boldsymbol{a}}^-(u) = \inf\{\sum_i a_i\delta_i : \boldsymbol{\delta} \in K_n(u)\},$$

then

$$\mathbb{P}_{\boldsymbol{\theta}}\left(\forall \boldsymbol{a} \in \mathcal{A}^d : \sum_i a_i\delta_i \in \bigcap_{n=1}^{\infty} [l_{n,a}^-(u), l_{n,a}^+(u)]\right) \geq 1 - u.$$

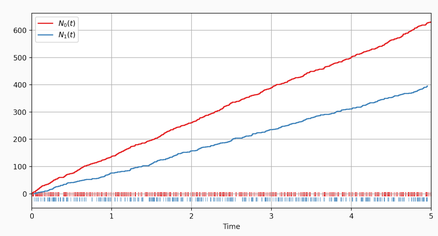
The upper bound $l_{n,a}^+(u)$ is the solution to the following convex optimization

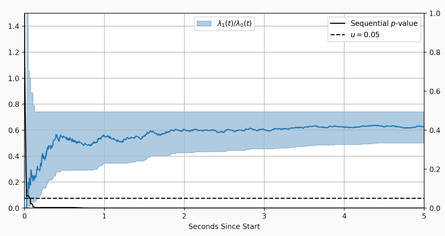
$$\begin{aligned} \max \quad & \sum_i a_i\delta_i \\ \text{s.t.} \quad & c \leq \sum_i S_i^n \left(\delta_i + \log \rho_i - \log \sum_j \rho_j e^{\delta_j}\right) \end{aligned} \tag{9}$$

where $c = \log \text{Beta}(\boldsymbol{\alpha}_0 + \boldsymbol{S}_n) - \log \text{Beta}(\boldsymbol{\alpha}_0) + \log u$. Convexity follows from the log-sum-exponential function. The lower bound $l_{n,a}^-(u)$ is the solution to the corresponding minimization problem. The confidence sequence for the contrast $\exp(\delta_1 - \delta_0)$ is visualized in right figure for the signup funnel experiment.

Case Study: Drop in SPS Detected

The following example is taken from a software canary experiment testing a new version of the Netflix application that contained a serious bug, preventing approximately 60% of all devices from streaming. Using the methodology for time inhomogeneous Poisson processes, the bug was detected in less than 1 second. The left figure shows the raw data, while the right figure shows the confidence sequence on $\exp(\delta_1 - \delta_0)$.





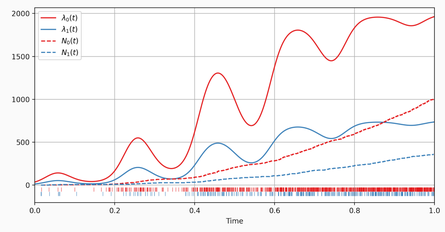
Sequential Tests of Time-Inhomogeneous Poisson Processes

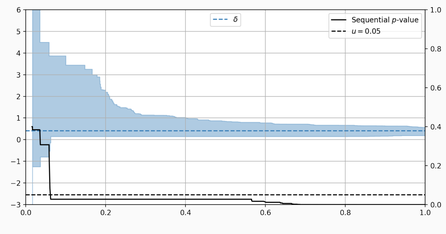
In Software deployments, developers carefully monitor the rate at which "successful play starts" (SPS) are received. Essentially, whenever the users application begins a successful playback of a requested title, an SPS event is sent to Netflix. If the rate of these events fall when testing the new software, then there is some error preventing titles from starting. In addition to SPS events, errors along with error codes are logged. This data forms a point process in time. Moreover, the instantaneous rate is not constant, because of time-varying usage patterns of users. We seek a sequential test for differences in time inhomogeneous Poisson point processes.

Consider d inhomogeneous Poisson point processes with intensity functions $\lambda_i(t) = \rho_i e^{\delta_i}\lambda(t)$ for $i \in \{1, 2, \dots, d\}$. Let each point produced by process i be marked with the corresponding process index i. At any time t, such as immediately after the previous point, the probability that the next point has mark i is given by

$$\theta_i = \frac{\rho_i e^{\delta_i}}{\sum_{j=1}^d \rho_j e^{\delta_j}}. \tag{10}$$

This gives the probability that the next point in time is from process i. This states that the sequence of marks can be considered a sequence of Multinomial$(1, \boldsymbol{\theta})$ random variables, allowing the sequential multinomial test to perform inference on $\boldsymbol{\delta}$. The right figure shows a confidence sequence for $\delta_1 - \delta_0$ in addition to a sequential p-value for testing $\delta_1 = \delta_0$.





Always-Valid Inference on Distributions

One of the most critical metrics for Netflix to monitor between software versions is *PlayDelay*. An important key indicator of the streaming quality at Netflix, it measures the time taken for a title to start once the user has hit the play button. Differences in the mean of the distribution of *PlayDelay* are not of primary interest. Instead, it is increases in the right tail of the distribution which is more concerning. Consider two users, one with a fast internet connection, the other with a slow internet connection, and consider the effect of *PlayDelay* on their satisfaction with the service. The former user is less affected by increases in *PlayDelay*, as the resulting values are likely still small and manageable, but the latter user is more affected, as their values were already high to begin with. As streaming performance is correlated with user dissatisfaction, increases in *PlayDelay* increase the risk of the latter user churning more than they do for the former. In other words, increases in *PlayDelay* are not nearly as important for the average user as they are for users already at risk. Large values of *PlayDelay*, even if infrequent, are unacceptable and considered a severe performance regression by our engineering teams.

We perform inference on the distribution of *PlayDelay* using the confidence sequences of **Howard and Ramdas[1]**. The following recounts their results and demonstrates an application to testing *PlayDelay* at Netflix:

$$\begin{aligned} \mathbb{P}[F_n^l(\alpha, x) \leq F(x) \leq F_n^u(\alpha, x)\, \forall x \in \mathcal{X}\, \forall n \in \mathbb{N}] \geq 1 - \alpha, \\ \mathbb{P}[Q_n^l(\alpha, p) \leq Q(p) \leq Q_n^u(\alpha, p)\, \forall p \in [0, 1]\, \forall n \in \mathbb{N}] \geq 1 - \alpha, \end{aligned} \tag{11}$$

with

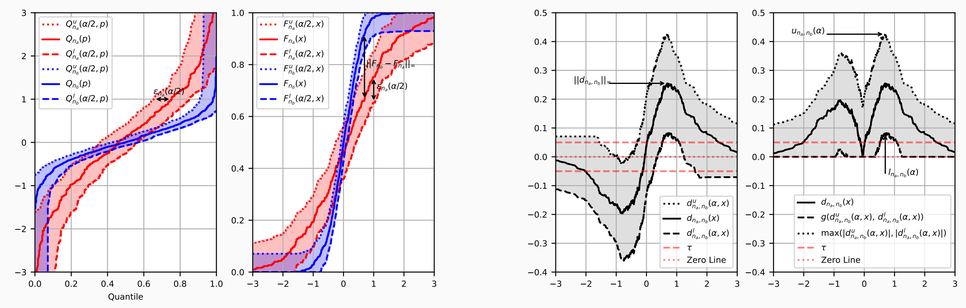
$$F_n^u(\alpha, x) = \min(1, F_n(x) + \epsilon_n(\alpha)) \qquad F_n^l(\alpha, x) = \max(0, F_n(x) - \epsilon_n(\alpha)), \tag{12}$$
$$Q_n^u(\alpha, p) = Q_n(p + \epsilon_n(\alpha)) \qquad Q_n^l(\alpha, p) = Q_n^{\rightarrow}(p - \epsilon_n(\alpha)), \tag{13}$$
$$\epsilon_n(\alpha) = 0.85\sqrt{\frac{\log\log(en) + 0.8\log(1612/\alpha)}{n}}. \tag{14}$$
$$\tag{15}$$

where $F_n(x) = \frac{1}{n}\sum_{i=1}^n 1[x_i \leq x]$ is the empirical distribution function, $Q_n(p) = \sup\{x \in \mathcal{X} : F_n(x) \leq p\}$ is the upper empirical quantile function and $Q_n^{\rightarrow}(p) = \sup\{x \in \mathcal{X} : F_n(x) < p\}$ is the lower empirical quantile function.

Let the difference between distribution and empirical distribution functions be denoted $d_{a,b} := F_b - F_a$ and $d_{n_a,n_b} := F_{n_b} - F_{n_a}$, respectively, then a confidence sequence on the difference between distribution functions can be obtained with

$$\mathbb{P}[d_{n_a,n_b}^l(\alpha, x) \leq d_{a,b}(x) \leq d_{n_a,n_b}^u(\alpha, x) \forall x \in \mathcal{X}\, \forall (n_a, n_b) \in \mathbb{N} \times \mathbb{N}] \geq 1 - \alpha, \tag{16}$$

where $d_{n_a,n_b}^u(\alpha, x) = F_{n_b}^u(\alpha/2, x) - F_{n_a}^l(\alpha/2, x)$ and $d_{n_a,n_b}^l(\alpha, x) = F_{n_b}^l(\alpha/2, x) - F_{n_a}^u(\alpha/2, x)$. The figures below show the confidence bands on the quantile functions, distribution functions, difference in distribution functions and absolte difference in distribution functions.



[1] Steven R. Howard and Aaditya Ramdas. 2021. Sequential Estimation of quantiles with applications to A/B-testing and best-arm identification. arXiv:1906.09712 [math.ST]

Case Study: Increase in PlayDelay Detected

In the following case study, we show how the sequential methodology successfully detected a performance regression in *PlayDelay* and prevented the release candidate from reaching the production environment. The data is taken from a canary test in which the behavior of the client application was modified. The null hypothesis was that observations of *PlayDelay* in the release candidate should be stochastically less than or equal to observations in the existing version ($H_0 : F_b \geq F_a$). The first figure shows the sequential p-value for this hypothesis as a function of time since the beginning of the canary test. The p-value falls below 0.01 after approximately 65 seconds.