Causal Impact for App Store Analysis

William Martin
University College London, United Kingdom
w.martin@ucl.ac.uk

ABSTRACT
App developers naturally want to know which of their releases are successful and which are unsuccessful. Such information can help with release planning and requirements prioritisation and elicitation. To address this problem, I performed causal analysis on 52 weeks of popular app releases from Google Play and Windows Phone Store. The results reveal properties of successful releases in multiple app stores, and showcase causal analysis as a useful technique for developers seeking to know more about their software releases.

Keywords
App Store Mining and Analysis, Causal Impact

1. INTRODUCTION
App developers are often motivated to adopt high frequency release strategies [3, 6, 7, 8, 10, 12, 16, 17]. Updates can be made for reasons beneficial to users such as for fixes, improvements and new features, and they can also be made for reasons not beneficial to users, such as updating advertisement libraries [8]. Sometimes updates are made simply to try and stimulate an app’s relative performance in the store [3], although high code churn can lead to decreased ratings [7]. Popular apps have been found to have high update frequencies, which do not correlate with their ratings [17].

Developers are privy to information about their own apps, such as sales and cash flow; as outsiders we do not have this information. However, app stores provide a method of measuring app performance, in the form of download ranks, ratings and reviews. I recorded metadata about the most popular apps in Google Play and Windows Phone Store over a 52 week period, in order to track their relative performance. Causal Impact Analysis was performed on 1,547 releases to identify those which may have caused significant changes in app performance.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM Student Research Competition (SRC) '16 Austin, Texas USA
© 2016 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-4205-6/16/05.
DOI: http://dx.doi.org/10.1145/2889160.2891033

2. CAUSAL IMPACT ANALYSIS
Causal inference [11, 13] is a technique used for exploring events of significant impact on time series data. It is used in economic forecasting, and also, recently, in software defect prediction [4, 5, 26]. Causal Impact Analysis [1, 2] is a form of causal inference, that works by training on a pre-event data vector in order to make a counter-factual prediction. This prediction tells us the most likely course of the vector after the event, from which we can determine whether a significant change has occurred in the observed data vector.

The counter-factual predictor has three components: i) local trend, which works by sampling a normal distribution for noise between time points, based on the variance in the pre-event time period; ii) (optional) seasonal trend, which applies a repeating bias that sums to zero over its time period; iii) control trend, which applies coefficients to a set of data vectors from external unrelated objects, and serves to account for global variance in the model.

Fig. 1 shows an example significant release, identified by my analysis, for the “Carp Fishing Simulator”.

3. STUDY DESIGN
I collected information on the rating, download rank, number of ratings and number of ratings in the last week from 307 Google Play apps and 726 Windows Phone Store apps over 52 weeks.

I applied Causal Impact Analysis using the CausalImpact framework, with no seasonal trend, and using the set of apps with no releases over the 52 week time period for the control trend. This set of apps consists of 97 apps in the Google Play dataset, and 397 apps in the Windows Phone Store dataset.
The set of apps with potentially impactful releases consisted of 210 Google Play and 539 Windows Phone apps; the set of releases for which there was sufficient prior and posterior information to perform causal impact analysis was 1,547.

I set a 90% confidence interval on results, meaning that only deviations with a less than 1% chance of being observed, if there had been no impact, were deemed significant. A separate experiment needed to be run for each performance metric and release, in order to train a separate predictor in each case. Each experiment then returned a p-value indicating the likelihood that the cumulative difference between the observed vector and the prediction could have occurred.

4. RESULTS

After applying causal impact analysis to releases in Google Play and Windows Phone Store, 301 out of 754 releases (40%) were found to be impactful in Google Play, and 437 out of 793 (55%) were found to be impactful in Windows Phone Store. Each impactful release was found to affect at least one of the four performance metrics positively or negatively. In both stores, approximately half of the releases impacted rating in some way: 20% positively impacting rating and 30% negatively impacting rating.

I grouped the impactful releases together in each store, in order to compare against releases which were not found to significantly impact app performance. I also compared the set of impactful releases that positively affected rating, against those that negatively impacted rating, in order to identify ways in which developers might be able to increase their ratings. The following candidate causes were considered, for the differences between these groups of releases: price, day of release, and length of release description text.

As shown in Fig. 2, across both stores, impactful releases were more likely to positively impact rating if they were more expensive. The length of release text also plays a factor: releases with longer (presumably more descriptive) release text, are more likely to be impactful, and to positively impact rating. Fig. 3 shows histograms for the releases on each day of the week, comparing impactful against non-impactful releases. We can see that releases are more likely to be impactful if released between Saturday and Tuesday in both app stores.

More results can be found in the technical report [15].

5. RELATED WORK

Past studies on app store release planning have looked into day of release [6, 10], potential reasons for updates [3, 8] and update frequency [17]. I built on this work and looked at releases which significantly impact app performance in their app stores, using empirical data mined from app stores as is commonly used in app store analysis [9, 14, 18, 22, 24, 25]. Previous studies have compared Blackberry and Android stores [23], and compared multiple Android stores [20, 21]; I compared properties between releases in Google Play and Windows Phone Stores. Past studies such as the one by Ruiz et al. [19] have used longitudinal data to monitor changes in rating; this study monitors such metrics, and looks for releases which may have caused significant deviations to them.

6. THREATS TO VALIDITY

In order to imply a causal relationship between releases and subsequent app performance, we need to apply a very large assumption indeed: that no external events may have caused the observed changes to an app’s relative performance. Clearly, this assumption cannot be made, and so we cannot assume true causality between releases and the changes observed. However, we can use causal impact analysis to identify those releases after which a performance metric changes significantly; then, by grouping such ‘impactful’ releases together, we can ameliorate risk of external factors as cause for the observed changes, and can identify useful properties and heuristics for developers.

7. CONCLUSIONS AND FUTURE WORK

I have collected app metadata for the consistently most popular apps in Google Play and Windows Phone Store over a 52 week period. By performing causal impact analysis on individual releases, I have identified a subset of releases after which their app’s performance changed significantly. I have grouped together these ‘significant releases’ in order to identify candidate causes for their impact, and found that price, day of release and descriptions of release content are factors in a release’s likelihood to be impactful, and to positively impact app rating.

This study has shown that causal impact analysis is a useful method for identifying individual releases which may have affected an app’s performance. This is particularly useful for developers, who may wish to run the tool on their own releases to work out specific successes or failures, in order to adapt their releasing or requirements development approach going forward.
8. REFERENCES


