Modeling the DAU of Social Applications

Michael Twardos

Abstract

Internet applications such as social games grow and die according to the actions of their users. In this paper, we define metrics that can be used to quantify user behavior to predict DAU over time.

Introduction

The number of active users on a given day (DAU) is the biggest indicator of an applications success. Other metrics, such as average revenue per user, ARPU, are derivations of DAU. On Facebook, there are currently 47 applications that have more than a million DAU. What characteristics of these applications make them successful. What metrics can help understand the rise an fall of an application's DAU. In what ways are DAUs predictable given this knowledge?

Power Laws

Quantifying user behavior means defining the probability a user performs a given action. For example, we can write the probability that a user plays space invaders d days after first trying it out. After likely being most interested when they first try the game, they may loose interest and their probability of returning decreases. A general function that decays, but never quite goes to zero that can describe such behavior is

$$p(t) = \frac{\alpha}{t^n} + C \tag{1}$$

where t is usually in the units of days. This type of function is often referred to as a power law function since the variable t is raised to the power we call n. The value α is a multiplier that simply increases or decreases the probability equally on every following day and *C* is a fixed offset. In the surprise of many, power laws, as opposed to other functional forms such as exponential decay, describe the likelihood of activity of a cohort for a given application. Figure 1 shows two power law distributions on log log axes.

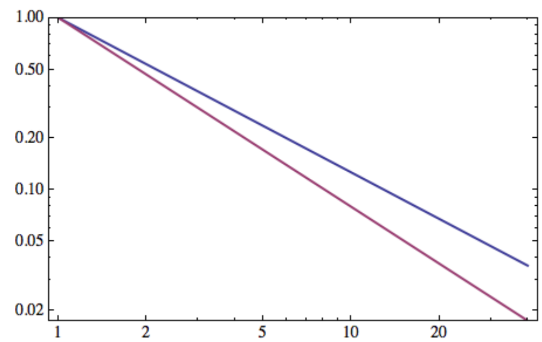

Figure 1: Two power law distributions, red is for $n = 1.1$ (convergent) and blue is for $n = 0.9$ (divergent) on a log-log plot. On these plots, the function is a straight line with a slope of n.

Modeling Active Users

Overview

We can use the mathematics discussed in the previous section to identify metrics that influence the DAU. DAU on a given day is determined by two things: how many users joined that day (for the first time) and how many users returned on that day given that joined previously. We will refer to the number of users joining on the, Tth day as $N(T)$. The variable T is in absolute time, so that N(1) is the number of users joining on the first day the application was released and N(2) is the number of users born on the second day, etc. Next, we will define a metric $R(t)$ that defines how many users returned to the application on the t^{th} day of a their life. For example, $R(2)$ is the probability the user returned and were active one day after they joined. $R(1)$ is the probability that users were active when they were born, which, depending on definition is likely to be 1 (Users only join an application because they were active). Finally, we define a metric $V(t)$ that defines how many users a given user invites on the t^{th} day of their life. In the following sections, we discuss how to use these metrics, N, R, V, to predict changes in the DAU.

Virality

Virality is a significant way for an application to gain more users. It occurs through a user's viral actions such as feed stories published or requests sent to friends. Virality of a given collection of users is the number of total viral offspring they have, divided by the number of users in the initial cohort. For example, if 100 users invite an additional 50 users, the viral coefficient of that initial user base is 0.5. However, not all users join an application directly through virality, at least a virality that can be measured. In the case of a user clicking on a feed story, it is possible to actually measure this viral action (embedding the source of the parent id in the links that their children click on). In other cases, users may have heard of an application from their friends and then searched for it before entering organically.

The virality of a user always has the potential to increase over time: there is always the possibility that a user can invite more users. Therefore, virality is always within the context of some amount of time, so we can write it as $V(d) = \alpha/t^n$ as in our previous discussion. Virality is often largest on a user's first day of life and trails off due to both decrease likelihood of retention and the saturation of user's own social network.

Virality, with the addition of users that were obtained through organic means or ads, determines the amount of new users joining an application. Given an initial set of users, we can predict how many users will join in the following days by iterating on virality for subsequent generations. The users acquired on the dth day can be written as:

$$N(d) = \sum_{t=1}^{d-1} N(t)V(d-t) + Organic \tag{2}$$

In other words if we want to determine how many users will join on the third day of an application's life, given that we know that $N(1) = 1000$ and $N(2) = 2000$ and having determined virality, $V(t)$,then $N(3) = V(1)(2000)I + V(2)(1000)$. Figure 2 show a more complicated example. In this figure $\alpha = 0.2$, $n = -1$ and the initial (seed) population is $N(1) = 1000$. In this scenario, user number drops significantly before growing back to 7 thousand new users per day by the 50^{th} day.

Saturation

In the previous section, we have assumed that there is an unlimited number of users that can potentially join the application. This is untrue for at least two reasons: 1) The size of the platform: currently there are less than 1 billion people with internet access and only 300 million users that have ever used facebook. 2) The size of the audience: likely to be significantly less than the size of the platform (for example if the app targets a certain demographic). Saturation can be modelled as a decrease in potential virality by the fraction that acquired users make up with respect to the estimated total audience size. In other words, there are some users that another user would potentially invite have already been invited (dues to finite size effects). The saturation factor can be written as:

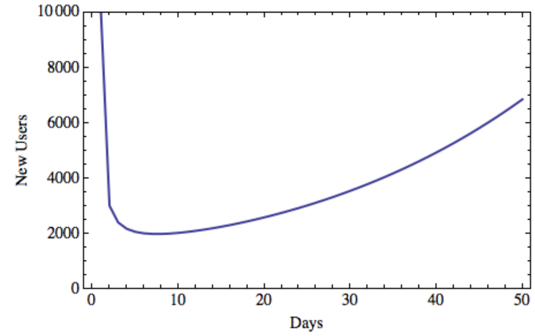

Figure 2: Suppose on the first day, 1000 users are acquired: $N(1) = 1000$. If our virality is such that $\alpha = 0.2$ and $n = -1$ then we can iterate how many new users will join per day based on those metrics.

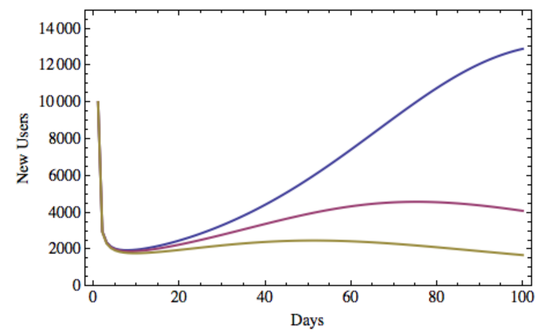

Figure 3: For a virality of $\alpha = 0.2$ and $n = -1$ with three different saturation levels. The blue corresponds to $S = 3M$, red is for $S = 1M$ and gold is $S = 500K$

$$N(d) = \sum_{t=1}^{d-1} \left(\frac{S - \sum_{q=1}^{d-1} N(q)}{S} \right) N(t)V(d-t) \tag{3}$$

where S is the maximum threshold size. Although this is a simple formula, we will show later to what extent this models real data. In figure 3, the growth of new users is shown for the same virality and different levels of saturation. A general increase in number of new users followed by a peak and decrease is observed.

Retention

In this section, we define retention: how many users will return to the application, given that they previously joined. Retention is denoted by the variable R and is a function of the users age, t as in $R(t)$. It is usually as a function of days, as in *What is the probability that a user returns t days from the day they were born?*. Retention can be measured at different times: the one day, one week, one month, one year. Figure 4 shows two hypothetical retention curves, one with a better short term retention (high α) and one with a better long term (high n). Figure 5 shows how a constant term in

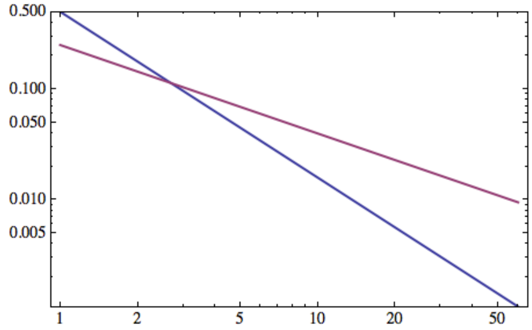

Figure 4: Retention rates of two different applications on a log-log plot. The blue retention is has an $n = 1$ and an $\alpha = 0.5$ while the red curve has an $\alpha = 0.25$ and a $n = 1/2$. While for users who are active in the first few days of their life, the blue curve has a higher probability to return, the red curve decays much slower and is better at retaining users overall.

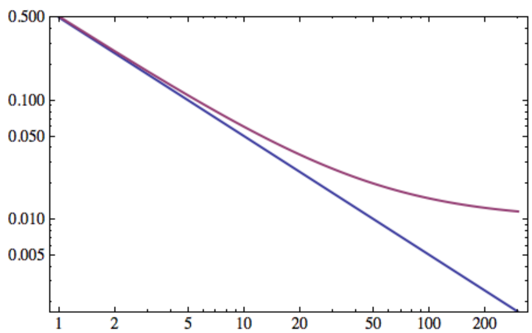

Figure 5: Retention curves both having an $n = 1$ and an $\alpha = 0.5$. The red curve has a $C = 0.01$ providing a steady retention rate at long times.

the retention curve indicates a steady consistent user base changes the overall retention rate compared to no constant. Attributes such as $alpha$ and C that have small effects on the retention curve may potentially have large impact on the number of DAU of an application as we will see in the next section.

Given that we know the number of users that joined, $N(t)$ and the probability that any of those users return d days later is $R(t + d)$ we can write down the number of users that will return today $A(t)$ as

$$A(t) = \sum_{d=1}^{t-1} N(d)R(t - d) \quad (4)$$

For a simple example if that we know that $N(1) = 1000$ and $N(2) = 2000$ and we also know that $R(t) = 0.5$ regardless of the age t of the user, then we can know that on the third day, 1500 users will return.

Modelling DAU

Given that we can write down the number of users joining an application and returning to an application every day, we

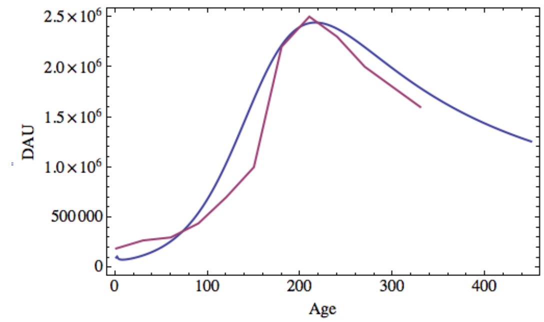

Figure 6: The DAU of Mindjolt games (apps.facebook.com/mindjolt) during 2009 in red, while the blue is an approximate fit to the growth using static variables for retention and virality. Virality has an $n = -1$ and an $\alpha = 0.25$ and a saturation, $S = 110000000$. Retention has an $n = 0.8$ and an $\alpha = 0.25$ and a constant offset $C = 0.0015$.

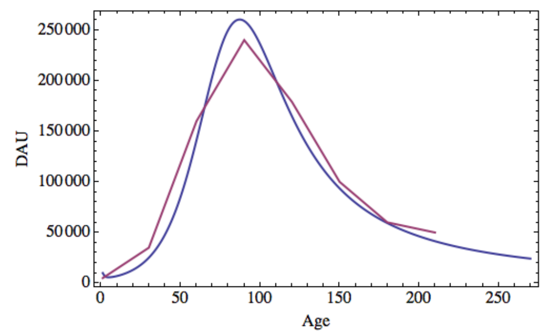

Figure 7: The DAU of Slide's SPP Ranch game (apps.facebook.com/sppranch) during 2010 (from January to August) in red, while the blue is an approximate fit to the growth using static variables for retention and virality. Virality has an $n = -1$ and an $\alpha = 0.37$ and a saturation, $S = 13000000$. Retention has an $n = -1.2$ and an $\alpha = 0.36$ and no constant offset $C = 0$.

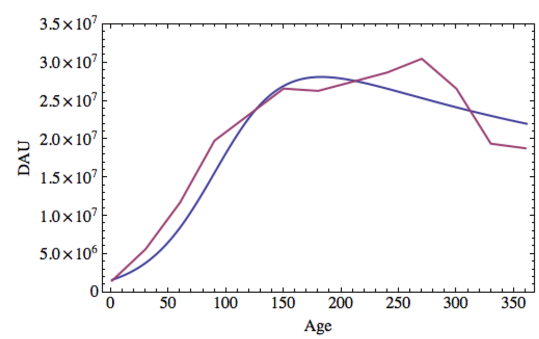

Figure 8: The DAU of Zynga's Farmville game (apps.facebook.com/farmville) between June 2009 and June 2010 in red, while the blue is an approximate fit to the growth using static variables for retention and virality. Virality has an $n = -0.9$ and an $\alpha = 0.25$ and a saturation, $S = 400000000$. Retention has an $n = -0.5$ and an $\alpha = 0.75$ and a constant offset $C = 0.001$.

can write down an equation for the DAU on a given day:

$$DAU(d) = \sum_{t=1}^{d-1} N(t)R(d-t) + \sum_{t=1}^{d-1} \left(\frac{S - \sum\limits_{q=1}^{d-1} N(q)}{S} \right) N(t)V(d-t) \tag{5}$$

If the retention and virality of an application are known and can be assumed to not change , they can be plugged into the equation to predict DAU at future times. The following figures show the DAU of three facebook applications: Mindjolt Games, Slide's SPP Ranch and Zynga's Farmville. In each of these cases, the chosen values for virality and retention are only speculation. Only the companies who actually track these numbers can know for sure how well these guesses were made. At the same time, these approximations are suprisingly close to the actual values. Each fit corresponds well to the viral growth these apps experienced early in their lifetime and the plateau and eventual decay of users at later times. It is interesting to not that each of these apps failed to continue growth or maintain a consistent active user over the course of even a year. This is an indicator that without consistent novelty or a game to continually reinvent itself, holding on to the users acquired over time is a very difficult task.

Conclusion

The popularity of social games are driven by the simple metrics of acquiring and maintaining users (virality andf retention). In this paper we showed how to use these metrics to model DAU over time. We showed how the correct parameters entered into a model can be used to follow closely the DAU of real applications for Zynga, Slide and Mindjolt. In doing so, we are able to estimate the retention and virality necessary to perform the feat that they did. In each case however, the DAU rises and falls due to a combination of a trailing virality (due to saturation) and a decreasing retention identifying the inability for an application to hold on to the users that they acquired. For meager retention curves, that fall very close to zero after only a few weeks or months is unsustainable: no massive viral acquisition of users can make up for the fact that the app will loose them soon after. At the same time, the application will never be able to acquire enough users to be successful if the virality is substandard. The good news is that virality and retention often go hand in hand: a quality application often acquires and retains users equally well.