Optimal Impression Allocation

Michael Twardos and Adam Cunha

Introduction

Video games come in all shapes and sizes. With the advent of the internet and the transition to more time spent on social networks, new types of video games incorporating social elements and time management have grown in popularity. In addition to the more standard game mechanics involved in the development process, these games also include viral features for users to share their experiences with friends through invites and challenges. This shift is one of the reasons for an increased emphasis on analytics in the game development process and maintenance. Recording the behaviors of users such as the levels they complete, the achievements they earn as well as the feed stories they publish and the friends they invite are a central component of many successful social games.

Gaming platforms that host large amounts of content provide ways for players to find new games and explore what games their friends or the rest of the social network find best. In this case, ”content management” is the challenge: Given that the platform has limited opportunity to showcase what content the user may like, what types of content does it show the user? Similar to other content platforms including social and mobile applications, music, photo and video directories, gaming platforms display similar collections. The most popular, newest, most popular newest, highest rated, editor's choice or similar / recommended for you are some of the ”content lists” seen most often across most content platforms. Many of these lists are logical: the most popular content is the content chosen by the largest audience and therefore the most likely to be enjoyed by the next user. Content that is ”similar” (users who chose A also often chose B) also makes sense. Editor's choice is for the quality content not reflected in the other lists.

In this paper, we will explore a mathematical framework for content lists on social gaming platforms. Specifically, we attempt to quantify to what extent are these lists optimal. Also, we hope to provide a framework or a methodology based on observable quantities of user behavior to display optimal content lists.

Definitions

To begin with, we will define some variables. The amount a given game is shown is often referred to as impressions. We define I_q as the relative probability that the q_{th} game is shown. Since the total probability must add to one and given that we have a platform that contains N games we can write

$$\sum_{q=1}^{N} I_q = 1 \tag{1}$$

If H is the total number of impressions we are able to provide then we can write HI_q as the total impressions we provide for game q.

For each game we are able to make a measure of events, X (X_q is a measure of X on the q_{th} game). Also suppose that this measure X is a function of the impressions I that we provide for it (as in $X(I)$). One common example of a metric that depends on the impressions provided for a game is the ”clicks” that take a user from the impressed game to the game itself. The relationship between the users clicking and the impressions provided is the ”click through rate” (referred to as ctr for short) which describes the number of users clicking (or total clicks) per impression provided. For example, if we impressed a game 1000 times on a set of users and 200 users clicked, it would have a ”click per impression” (CPI) value of 0.2.

One point to note, before we go further, is the issue of errors or uncertainty in measuring X_q and XPI_q for all games. The error scales inversely proportional to the ”sample size”. In this case, the sample size may include the amount of impressions we provide for a game as well as the number of users that click on an impression to perform some additional analysis. In this case, errors become significant when an active user base is small and/or the number of games is large. For the purposes of this discussion, we assume that we can measure X_q and XPI_q exactly. However a proper treatment of error analysis and error propagation is necessary and requires additional attention.

Continuing, suppose we measure that a game q has a CPI value of CPI_q. Then HI_qCPI_q is the total number of users clicking on that game. Total number of users clicking across all games would be

$$H \sum_{q=1}^{N} I_q CPI_q = \text{total users clicking} \tag{2}$$

The astute reader may notice that this is long form for the vector product between the impression vector, **I** and the clickers per impression vector, **CPI**. In this case, each component of the vectors is another game so we will refer to these as "game vectors".

Continuing, suppose we are able to follow these users after they have clicked on the "game ad". We can record how many games they play, how many feed stories they publish and how many virtual goods they buy, etc. In this sense, we can make any measure on a "per impression" basis. We refer to this generic measure per impression as **XPI**. For example, suppose we impressed a game 1000 times and 200 users clicked on it. The users on that day played 600 total games (an average gameplay per user of 3). Therefore, we would measure this game to have a characteristic "gameplays per impression" , **GPI** of 0.6. This means that for every impression that we provide for this game, we can expect to generate 0.6 gameplays. This metric is clearly combination of the ctr for that game (clicks (or users) per impression) and gameplays per user. In this sense, every metric that can be determined on a per user basis for a game can be translated into a per impression basis using the **CPI** vector.

Game Vectors

In figure 1, we show a graphical representation of a fictional XPI vector for a fictional platform with N=100 games.

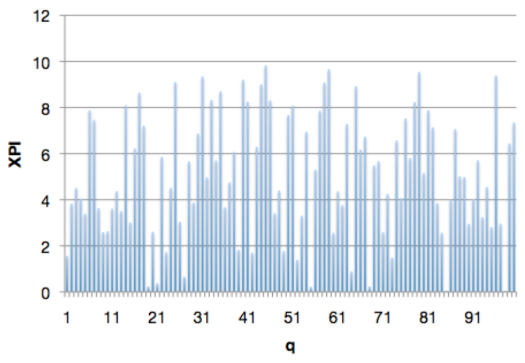

Figure 1: A list of XPI values for 100 games unsorted.

A convenient way to write this vector is by sorting the components from greatest to smallest as shown in Figure 2.

The sorted representation of game vectors provides a better understanding of the variety of content quality in terms of the XPI metric. (Mention entropy and Gini coefficient? to characterize distribution?) The shape of this distribution may be exponential, power law, linear or logarithmically descending in value, etc as shown in Figure 3. As we will see in the next section, the more this shape is skewed (the largest XPI values for the fewest games, as in a power law shape, the more important it is to impress these games correctly.

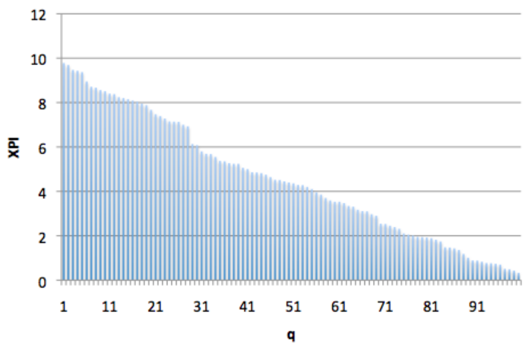

Figure 2: A list of XPI values sorted in descending value by XPI.

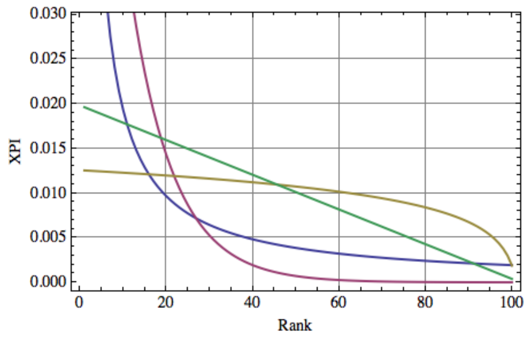

Figure 3: Examples of game vectors, **XPI**, with power law (blue) with an exponent of -1, exponential (red), linear (green) and logarithmic (yellow) shapes .

Optimal Impression Vectors

Using this vector terminology, we are now able to rephrase our question in a different way: given that we measure the vector **XPI** over all components (games) find **I** such that the product between these vectors is maximized. For the time being we assume that X_q is linear with I_q ($XPI_q(I)$ is constant). For low amounts of impressions, this is a good approximation. In this case, the solution is easy: The impressions should all be allocated for the game with the highest XPI. (Why? What is this called?) We refer to this optimal Impression vector as $\mathbf{I_1}$ and it can be written as

$$\mathbf{I_1} = \begin{cases} 1 & \text{if q = 1} \\ 0 & \text{if q != 1.} \end{cases} \tag{3}$$

In this case the amount of X we generate by using $\mathbf{I_1}$ is $H\mathbf{I_1}\mathbf{XPI}$ =$H{*}XPI_1$ In many instances, we have the opportunity to impress more than one game at the same time (say 5 slots in a "cross promotion" bar). Given g opportunities, we can write $\mathbf{I_x}$ as

$$\mathbf{I_x} = \begin{cases} \frac{1}{x} & \text{if q <= x} \\ 0 & \text{if q > x.} \end{cases} \tag{4}$$

There may be other reasons besides x slots to show x games. For example, there may be additional games that

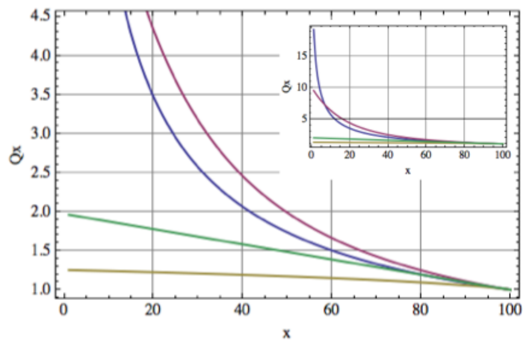

Figure 4: For the vectors shown in Fig 3, the values of Q_x as a function of increasing x. The power law (red) and exponential (blue) have significantly higher values for low values of x. For games distributed more evenly: linear (green) and logarithmic (yellow), there is a weaker dependence on impressing only the games with the highest XPI_q. The inset shows the same plot for the full range of values. As can be seen the power law has a Q_1 that reaches nearly 20, while the exponential reaches a Q_1 value of almost 15.

the platform is obligated to impress even though they do not have the best XPI_q. How much does the platform "loose" by impressing additional games? On one end of the spectrum, if we only impress the best game, we get $H\mathbf{I_1XPI}$. On the other hand, we can impress all N games the same amount and get $H\mathbf{I_NXPI}$. We can define a value Q_x defined as:

$$Q_x = \frac{\mathbf{I_xXPI}}{\mathbf{I_NXPI}} \tag{5}$$

that tells us how much better than impressing all games equally we will do by impressing the top x games of XPI. The results are shown in Figure 4 for different shapes of $\mathbf{XPI}$ vectors included in Figure 3. For $\mathbf{XPI}$ that are strongly skewed (power law and exponential), the results are most dramatic. Compared to impressing all games equally, we are able to generate over $10\times$ more X by choosing to impress games with the highest XPI_q.

Weighting the Impression Vector

Another approach to choosing impression allocation is to use an impression vector I_{wx} whose components are weighted by the value XPI_q of the q_{th} and only go up to the x_{th} game. In other words:

$$\mathbf{I_{wx}} = \begin{cases} L * XPI_q & \text{if q <= x} \\ 0 & \text{if q > g.} \end{cases} \tag{6}$$

where L is a normalization constant defined as

$$L = \frac{1}{\sum_{q=1}^{x} XPI_q} \tag{7}$$

In this case we can define a value R_wx defined as:

$$R_{wx} = \frac{\mathbf{I_{wx}XPI}}{\mathbf{I_xXPI}} \tag{8}$$

The values of R_{wx} are shown in Figure 5. For small values of x, values of R_{wx} are close to 1. In other words, weighting is not that important (not that meaningful). As we add more games to the list, the weighting value becomes more significant for the power law and exponentially distributed vectors. Using all 100 vectors and weighting them proportional to their XPI_q values as compared to random generates $5\times$ more X than weighting them equally.

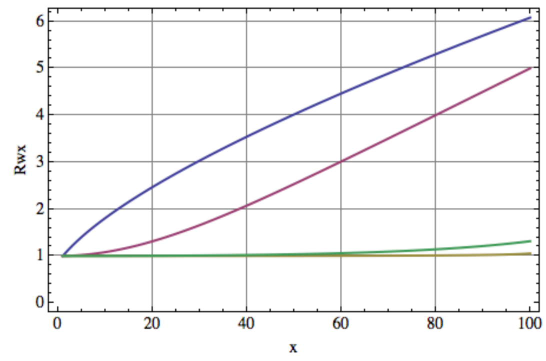

Figure 5: The ratio of weighting vectors with the top XPI according to their XPI_q values up the to the x_{th} value per weighting all components equal up to x. For $x = 1$ only the top game is impressed in all cases so all values converge to 1. For the exponential and power law curves, when impressing all $N = 100$ games, weighting the impressions of the q_th game XPI_q generates over $6\times$ as much return.

From our first result, $\mathbf{I_1}$, using any form of $\mathbf{I_{wx}}$ is not optimal. This vector is just used as an example of the situation when we are obligated to show many games, what type of loss or benefit would occur, given a different weight. Finally, we can weight our impression in a more extreme way by raising the components XPI_q to a power j as in

$$\mathbf{I_{jwx}} = \begin{cases} L_j * XPI_q^j & \text{if q <= x} \\ 0 & \text{if q > g.} \end{cases} \tag{9}$$

where L_j is a normalization constant defined as

$$L_j = \frac{1}{\sum_{q=1}^{x} XPI_q^j} \tag{10}$$

In this case, as we raise the power, j, we can get closer to the case in which we impress the top value, as in $\mathbf{I_1}$. Here, we can define another metric, S_j that tells us how much more X we generate by weighting each component proportional to XPI_q^j for all $x = N$ as

$$S_j = \frac{\mathbf{I_{jwx}XPI}}{\mathbf{I_{wx}XPI}} \tag{11}$$

The results of S_j are shown in Figure 7. Continuing to focus on the power law distribution (blue), as j approaches

∞, S_j goes to 3. To summarize, combining our results for the power law case with an exponent of -1, we get a $6\times$ increase when we weight all out $N = 100$ components proportional to their XPI_q value as opposed to equally. Then, as we raise the power j of the weight, we get closer to approximating the $\mathbf{I_1}$ vector and provides us with another $3\times$ increase. Together this gives us the initial result we achieved from determining Q_x for $x = 1$ which was around 20.

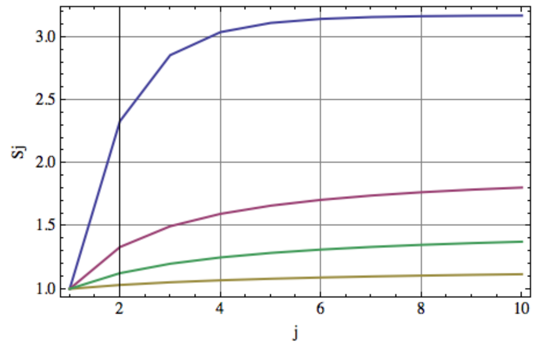

Figure 6: The increase in X given that we weight components all components asXPI_q^j, where $j \geq 1$. For the power law distribution, as j increases, we approach $3\times$ more.

Monkeys Choosing I

As a final sanity sheck, we can pose the question: "If we randomly generate impression vectors, would any of them do better than $\mathbf{I_1}$ or $\mathbf{I_x}$ or $\mathbf{I_{wx}}$?". In other words the q_{th} element of the impression vector is simply a random number, which is then normalized by the entire list of random numbers. We can write this random impression vector as $\mathbf{I_R}$. If lots of monkey can generate lots and lots of $\mathbf{I_R}$, how well can they possibly do. In particular, for a given $\mathbf{XPI}$, how much better can $\mathbf{I_R}$ perform than $\mathbf{I_N}$. If we define Z_R as

$$Z_R = \frac{\mathbf{I_R XPI}}{\mathbf{I_N XPI}} \tag{12}$$

then what do these probability distributions, $P(Z_R)$ look like? The results are shown in Figure 7. For any of the XPI vectors considered, there is no values of Z_R that are greater than 1.5. This suggests that our impression vectors considered earlier are indeed "rare" and not easy to generate by a brute force method.

Conclusion

In this discussion we explored several different aspects of impression allocation for a platform that has an inventory of N games. For a game, q, which has some observable quantity, X_q that has a linear relationship to the impressions provided $H * I_q$ and a characteristic XPI_q, how do we choose I_q optimally. After noting that the solution for a sorted game vector $\mathbf{XPI}$ is $\mathbf{I_1}$ we explored several variations that deviate from optimal. These deviations included adding more games to the impression vector, $\mathbf{I_x}$, adding weight to those games, $\mathbf{I_{wx}}$ and varying the weight of those games, $\mathbf{I_{jwx}}$.

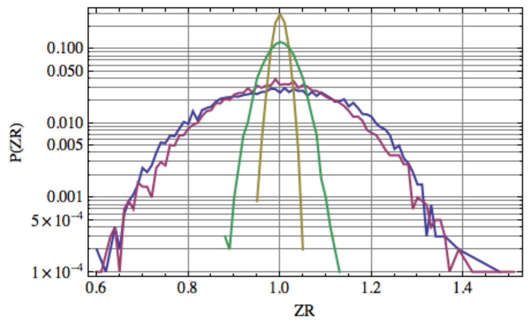

Figure 7: The probability distributions of Z_R for different XPI vectors shown in Figure 3. Even for the power law distribution (blue), there is less than a 1 in 10000 that a randomly weighted vector is more than $1.5\times$ the evenly distributed random vector I_N. We can use this to conclude that vectors such as I_1 and I_x which generate over $10\times$ more X than the I_N vector, are a very rare and optimal solution.

We measured the effect by comparing the results to the equal impression vector $\mathbf{I_N}$ by defining the quantities $\mathbf{Q_x}$, $\mathbf{R_{wx}}$, $\mathbf{S_{jwx}}$ for different "shapes" of $\mathbf{XPI}$ vectors. In some cases, particularly the heavy skewed distributions of power law and exponential vectors the results were most striking: over an order of magnitude difference in generating X when the optimum vector is chosen compared to $\mathbf{I_N}$. We did not explore how these results vary for different exponents of power law and exponentially distributed vectors, but we can guess that the more skewed the vectors are, the more drastic the results will be. Ideally a platform may contain many quality pieces of content, but realistically, content is likely to be skewed similarly to these heavy tailed distributions. In such a case, finding and impressing the best content takes even greater priority.

Sample References

Forthcoming Publication

Clancey, W. J. 1986a. Curious George.