Comparing Player Skill, Game Variants, and Learning Rates Using Survival Analysis

Aaron Isaksen aisaksen@nyu.edu
Andy Nealen nealen@nyu.edu
NYU Game Innovation Lab

Abstract

Game designers can use computer-aided game design methods to quantitatively compare player skill levels, different game variants, and learning rates, for the purpose of modeling how players will likely experience a game. We use Monte-Carlo simulation, hazard functions, and survival analysis to show how difficulty will quantitatively change throughout a game level as we vary skill, game parameters, and learning rates. We give a mathematical overview of survival analysis, present empirical data analyses of our player models for each game variant, and provide theoretical probability distributions for each game. This analysis shows the quantitative reasons why balancing a game for a wide range of player skill can be difficult; our player modeling provides tools for tuning this game balance. We also analyze the score distribution of over 175 million play sessions of a popular online Flappy Bird variant to demonstrate how learning effects can impact scores, implying that learning is crucial aspect of player modeling.

1 Introduction

We aim to help game designers better balance their games by quantitatively comparing players of different skill levels, comparing different game variants, and comparing how players experience a game after practice and learning from repeated plays. A designer can use game metrics (El-Nasr, Drachen, and Canossa 2013) to study and improve the game, but this requires the game to be publicly available, have significant numbers of players, and time to gather enough data – and therefore can't be used for rapid iteration in the early design process. Automated game testing (Nelson 2011), which simulates players of varying difficulty at high speeds, can be used for tuning a game. It's especially useful as designers and playtesters become experts at the game, and often forget how novices may experience it.

In this paper, we show how computer-aided game design, which we loosely define as a system where human designers work together with computers to co-create a game (Yannakakis, Liapis, and Alexopoulos 2014), can help us better understand our craft and provide knowledge to make better games. Specifically, we show how probabilities are a useful tool for understanding, modeling, and comparing player experience; in past work probability distributions have been used for modeling session times (Feng et al. 2005) and total play time (Bauckhage et al. 2012). Probability distributions tell us the likelihood that a player will achieve a specific score, and we can create these distributions by repeatedly playing a game and collecting the resulting score frequencies. As shown in Fig. 1a, differing player skill can give rise to significantly different score distributions. Isaksen et al (2015) show how to use exponential distributions for designing simple action games like Flappy Bird (Nguyen 2013) which do not change in difficulty when ignoring learning effects. In this paper, we explore games that increase in difficulty as a player progresses, and we make a first attempt at modeling learning effects, which we show has influence on the resulting scores. We also show how hazard functions (Rinne 2014), shown in Fig. 1b, can be used to theoretically and empirically model changes in difficulty for different players.

Through our analysis, we have discovered what we hope to be interesting and useful results for game designers. Firstly, we show with Monte Carlo simulation how to quantify how players of different skill levels will likely experience the same game variant in significantly different ways, giving the developer precise methods to accurately balance their game for varying skill. Secondly, these methods can help developers balance a game that has increasing levels of difficulty as the

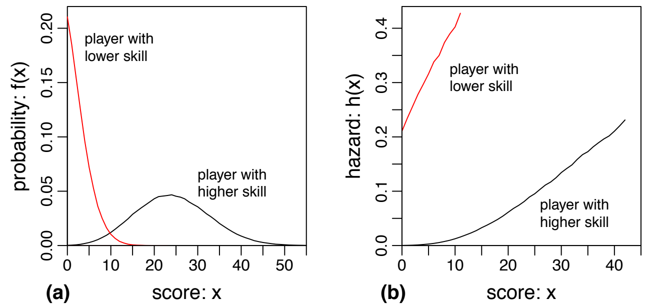

Figure 1: Two players with different skill playing the same game show uniquely different play experiences. (a) Resulting score probability distributions shows significantly different shapes. (b) Hazard functions, equivalent to a designer's "difficulty curve," show how the difficulty changes for each player.

player progresses. Finally, we show that learning effects can significantly impact the analysis of game data and should not be ignored when computationally modeling players.

We demonstrate these findings using a minimal one-button game (Nealen, Saltsman, and Boxerman 2011), where we record thousands of plays with an AI simulating human error due to motor skill imprecision. By varying game parameters, player skill, and introducing learning effects, we can generate score distribution data and then model empirically and theoretically how each change leads to varying difficulty.

To explain how the probability distributions predictably change, we use *survival analysis*, a branch of statistical modeling that helps predict how long in the future an event will occur (Lee and Wang 2013). It is used to study how long mechanical parts will last, the effectiveness of medical treatments, and for predicting human lifetimes, but to our knowledge is not commonly used by game designers. Here we use survival analysis to predict the likelihood of a player achieving a certain score. In most action games, the longer a player survives in a game, the higher their score, so we use score as a replacement for lifetime and leverage existing work on survival analysis. Although this paper focuses on measuring probabilities of achieving specific scores, one can use our approach to model difficulty as any non-decreasing factor, such as time played, coins collected, enemies killed, etc.

In Section 3, we give an short mathematical overview of survival analysis. A probability distribution $f(x)$, which we can obtain from actual or simulated game play, tells us how likely a player is to achieve a specific score x (Fig. 1a). From this, we can create a survival function $S(x)$, which tells us the likelihood that a player will reach a score $\geq x$. Finally, the hazard function $h(x)$ tells us how likely a player will die at a specific score x, given they have already survived up to that score (Fig. 1b). When a designer talks about the "difficulty curve" for a game, they are talking about a hazard function. We demonstrate how the hazard function is often a simpler way to compare different players and games.

We then simulate different games and players in Sec. 4, showing how varying different parameters of the game or player model leads to different probability distributions and hazard functions, and therefore player experiences. For each of these game types, we plot empirical probability distributions and derive hazard rates from the generated data, and then present a matching theoretical hazard function model.

Finally, we use our theory to analyze over 175 million plays of flappybird.io (McDonnell 2014). This dataset matches the type of probability distributions and hazards associated with our learning model, showing the likely presence of learning effects when humans repeatedly play a game. This has an important impact when designing games, so that the game remains interesting after repeated plays and does not become too simple for improving players.

2 Simulated Game Play

In order to explore how varying difficulty and player skill affects resulting game scores, we collect simulation data from a simple game that can be played by an AI using a model of human-like motor skill. The game, shown in Figure 2, is a minimal version of an infinite runner, and is designed to be easy and fast to simulate. In this model game, the player is standing above to a conveyor belt with an endless series of empty boxes. The player must tap a button to pack each box while it is under the player. If the player misses a box or taps when there is no box under them, they lose the game. The final score of the game is equal to the number of boxes successfully packed. We can expect that wider boxes would make the game easier since the player has a wider window for error, and faster belt speed would make the game harder since the player would have less time to react to each box.

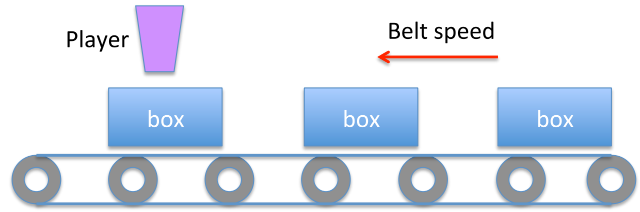

Figure 2: In our simulated game, the player earns a point by tapping a button once when the box is underneath. The game ends if a box is missed or the player taps when there is no box. In some variants, the belt speeds up after each point.

Isaksen, et al. (2015) showed that one can simulate human performance in simple action games using an AI which plans into the future and then adjusts the reaction time by a normal distribution. By increasing and decreasing the standard deviation of the normal distribution, we simulate less skilled (i.e. less accurate) and more skilled (i.e. more accurate) players. In our system, the AI tries to tap when the box is centered below the player, but we adjust the actual time the AI taps by a normal distribution representing the AI's skill.

We try different versions of the game, varying the single design parameter *belt speed*, and examine the impact on the distributions of scores that result from the simulation. We test versions where the belt speed is constant, as well as some that speed up as the player progresses. We also vary the modeled skill level of the AI, allowing us to (1) simulate novice and expert players and (2) simulate players that learn and improve each time they play. Each of these simulations results in data that shows a different probability distribution.

3 Overview of Survival Analysis

To understand how these simulations of games of different difficulty and players of different skill give rise to unique probability distributions, we need a mathematical foundation to model player performance. The simulation data we create are in the form of discrete scores, but the underlying distributions are easier to describe in the continuous domain.

3.1 Probability Distribution Functions

We begin by looking at the probability distribution function $f(x)$ for a game, which tells us the probability that the player will achieve a score of x on the next play. For example, if $f(5) = .20$ then the likelihood a player will achieve a score of 5 on the next play is 20%.

In this paper, for simplicity and clarity, *probability distribution function* (PDF) refers to a *probability density function* when using continuous probabilities and a *probability mass function* when using discrete probabilities. We explicitly denote *cumulative distribution functions* where needed.

In practice, we can create a discrete probability distribution by recording all of the scores on each play of the game, then summing up the frequency of each score, and dividing by the total number of plays. This is often recorded for each play when using analytics such as Google Analytics, and the frequency of each score is easily output by these systems. For games with a wide range of scores, or with gaps between scores, specific individual scores might have zero or very low frequencies. In this case, it could help to quantize the scores in a histogram.

3.2 Survival Functions

The *survival function* $S(x)$ tells us the probability that a player will still be alive given they have already reached a score of x on the current play. In other words, it describes the likelihood that a player will achieve a score $\geq x$. For example, if $S(10) = .25$ there is a 25% probability that the player will achieve a score of 10 or higher. It is closely related to more commonly known *cumulative distribution function* $F(x)$. The survival function is defined as:

$$S(x) \quad = \quad 1 - F(x) \quad = \quad 1 - \int_0^x f(s)ds \qquad (1)$$

$S(x) = 1$ when $x \leq 0$ because every player will at least achieve a score of 0, and $S(\infty) = 0$ because all players will eventually reach a termination state and receive a final score.

3.3 Hazard Functions

The hazard function $h(x)$ is useful for comparing probability distributions and understanding the difficulty of a game. Also called the hazard rate, it is defined as:

$$h(x) = \frac{f(x)}{S(x)} \qquad\qquad f(x) = h(x)S(x) \qquad (2)$$

The hazard function tells us the rate at which we should expect to fail, given we've already reached a specific score x. This is not the probability that we will fail at this exact score x, given by $f(x)$, but a conditional probability that the player has already survived to a score of x. For example, if $h(10) = .15$, this means that once the player gets to a score of 10, they now have a 15% chance of failing at this point.

The hazard function is especially useful when analyzing games because its directly related to how we as designers think about adjusting difficulty curves in a game. We aren't as concerned about the entire probability distribution as much as how difficult a game is at a specific section, assuming the player has already reached that point in the game. We give many visual examples of hazard functions in Section 4.

Therefore, game designers are effectively working on modifying the hazard function $h(x)$ so the difficulty curve feels good to players (Swink 2009). If we want to understand how these changes affect the resulting probability distribution $f(x)$ and survival function $S(x)$ for the game, we need a method to derive $f(x)$ and $S(x)$ from a given hazard $h(x)$.

First, we write the cumulative hazard function $H(x)$, which represents the total amount of risk (Cleves 2008) that a player has faced up to their current score x, as:

$$H(x) = \int_0^x h(u)du \qquad (3)$$

We then take the derivative of Eq. 1:

$$f(x) = -\frac{dS(x)}{dx} = -S'(x) \qquad (4)$$

Using Eq. 4, we rewrite Eq. 2 as $h(x) = -S'(x)/S(x)$. This is a first order differential equation with the following solution (Boyce, DiPrima, and Haines 1992):

$$S(x) = e^{-\int h(x)dx} = e^{-H(x)} \qquad (5)$$

By using Eqs. 4 and 5 we obtain the final relation:

$$f(x) = -\frac{dS(x)}{dx} = -\frac{d}{dx}e^{-H(x)} \qquad (6)$$

We can now derive a theoretical $f(x)$ and $S(x)$ from any theoretical $h(x)$, which helps us understand how changes in difficulty affect the resulting score distributions.

3.4 Working with Discrete Data

We use continuous distributions when building models, but since scores from our games are discrete values (that is we can receive a score of 1 or 2, but not 1.245), we work in the discrete domain when analyzing game data. We can create the discrete probability distribution $\dot{f}(x)$, discrete survival function $\dot{S}(x)$, and discrete hazard function $\dot{h}(x)$ as follows (a dot over each function signifies we are talking about the discrete domain).

First, we run the game N times and save each score in a vector $\mathbf{Z}$. We now create a histogram from this data with bin size of 1, and define $\dot{n}(x)$ as the number of scores in $\mathbf{Z}$ equal to x. We can now calculate these values from our data as:

$$\dot{f}(x) = \frac{\dot{n}(x)}{N} \qquad \dot{S}(x) = \sum_{s \geq x} \dot{f}(s) \qquad \dot{h}(x) = \frac{\dot{f}(x)}{\dot{S}(x)}$$

Because $\dot{S}(x)$ becomes very small as higher scores become less likely, $\dot{h}(x)$ is susceptible to noise. One option is to smooth the hazard function using techniques based on kernel smoothing (Muller and Wang 1994) or splines (Rosenberg 1995). For this paper, we simply do not plot $\dot{h}(x)$ for values of x where $\dot{S}(x) < \epsilon$, where ϵ ranges between 0.01 and 0.001 depending on how many samples were used to generate the plots. This avoids the noisiest parts of the hazard function, which are undersampled due to low probability.

4 Survival Analysis of Simulated Games

We now simulate variants of our experimental game, adjust parameters of the game and player model, and collect histograms of score data. This data is used to create a empirical discrete probability distribution $\dot{f}(x)$ for each game, and we calculate the empirical discrete hazard $\dot{h}(x)$ from $\dot{f}(x)$. We then show how each system can also be modeled with a matching theoretical hazard $h(x)$ which leads us to a theoretical probability distribution $f(x)$ that predicts the probabilities generated by our simulation.

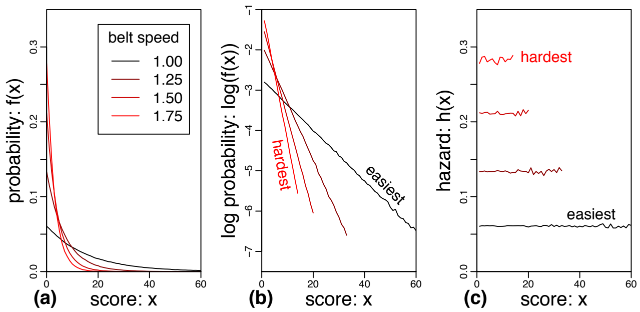

Figure 3: Data collected from simulated games with constant belt speed, constant skill level, and no learning effects. (a) Harder games have a higher probability of getting a lower score. (b) The log probabilities are linear, indicating an exponential distribution. (c) Constant hazard rates indicate constant difficulty.

4.1 Constant Difficulty: Exponential Distribution / Constant Hazard

We start by examining games that do not modify their parameters as the game progresses, and therefore have a constant difficulty if one ignores learning effects. Flappy Bird (Nguyen 2013) is an example of this type of game. We show with our simulation that these conditions lead to a constant hazard.

Using the score results from our simulated game, the data shown in Figure 3 shows empirical evidence of an exponential distribution and constant hazard rate when using a constant belt speed, constant skill level, and ignoring learning effects. In Fig. 3a we show the probabilities for 4 versions of the game, each with a different belt speed (increasing in speed from the black line to the red line). Exponential distributions become linear in log plots, so we can tell from Fig. 3b that the data indeed comes from the exponential distribution. The derived hazard rates, shown in Fig. 3c, also increase, indicating as expected that faster belts lead to a more difficult game. There is some noise in the hazard function as we are simulating human error using a stochastic process. This noise can be reduced with more iterations of the simulation. The harder games have shorter lines, because its unlikely a player will achieve higher scores in them.

Given the evidence for a constant hazard, we now theoretically model the constant hazard function $h(x) = \lambda$, which means that the player is equally likely to die at every moment in the game. Using Eqs. 2-6, we have:

$$\begin{array}{rcr} h(x) = & & \lambda \\ H(x) = & \int_0^x h(u)du = & \lambda x \\ S(x) = & e^{-H(x)} = & e^{-\lambda x} \\ f(x) = & h(x)S(x) = & \lambda e^{-\lambda x} \end{array}$$

Thus, a constant hazard rate (i.e. constant difficulty) leads to an exponential probability distribution. A more difficult game has a higher λ: a player is more likely to die after each point and has a lower likelihood of reaching a higher score.

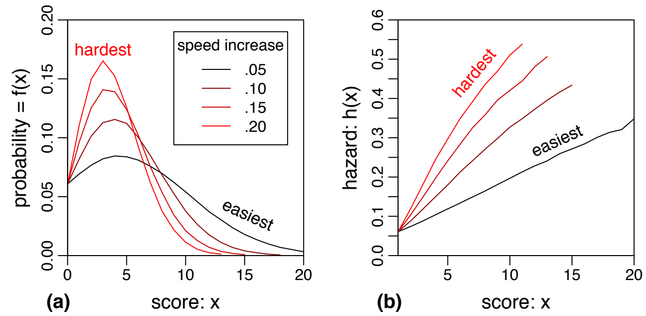

Figure 4: Data collected from simulated games with linearly increasing belt speed, constant skill, and no learning. After each point, the belt speed goes up by *speed increase*. (a) Slower speed increases are more likely to exhibit higher scores, following a shifted Rayleigh probability distribution. (b) Steeper hazard lines indicate faster difficulty increases.

4.2 Increasing Difficulty: Rayleigh Distribution / Linear Hazard

Most games do not exhibit constant difficulty, but instead increase in difficulty as the player gets further in to the game. Figure 4 shows four simulated games where we start with a shared belt speed and increase it by a fixed amount after each successfully packed box. The black line indicates the smallest increase in speed and the red line is the largest increase.

Fig. 4a shows the resulting empirical probability distribution for each variant. As expected, games with the slower belt speed increase show a higher likelihood of a higher scores. The initial game difficulty chosen for the experiment shows the nice design property that the player is more likely to achieve a score of around 3-5 than a score of 0, which means the player will likely experience some small success at the start (unlike the constant hazards described in Section 4.1).

In Fig. 4b we show the derived empirical hazard rate for each variant, which are approximately linear, although there is a slight curve downwards showing that the hazards aren't perfectly linear. Each line comes to the same point because each variant starts out with the same belt speed. Increasing the belt speed at a faster rate means the game gets more difficult more quickly, indicated by a steeper slope in the hazard plot.

We can theoretically model this with a linear hazard function $h(x) = a+bx$, where a defines the game's base difficulty and $b > 0$ defines the rate at which difficulty increases. Using Eqs. 2-6, we find the theoretical probability distribution:

$$\begin{array}{rcr} h(x) = & & a + bx \\ H(x) = & \int_0^x h(u)du = & ax + \frac{b}{2}x^2 \\ S(x) = & e^{-H(x)} = & e^{-ax-\frac{b}{2}x^2} \\ f(x) = & h(x)S(x) = & (a+bx)e^{-ax-\frac{b}{2}x^2} \end{array}$$

When $a = 0$ and $b = 1/\sigma^2$, this reduces to the well known one-parameter Rayleigh distribution $f(x) = \frac{x}{\sigma^2}e^{-x^2/2\sigma^2}$. Our data matches a two-parameter Rayleigh distribution with location parameter (due to the $a > 0$ constant term).

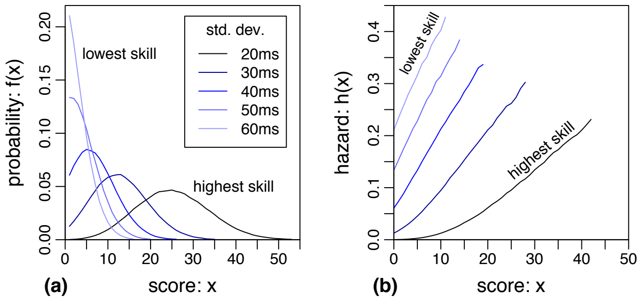

Figure 5: Data collected from simulated games with linear increasing difficulty. Smaller standard deviation models higher skill. (a) Skill greatly influences the shape of the empirical probability distribution. (b) Skill affects the y-intercept for the hazard rate, which causes the probability distribution in a. to shift and change shape.

4.3 Varying Skill Levels with Linear Hazards

We now explore how a single game variant with increasing belt speed can be experienced by players of different skill levels. In Figure 5 we simulate the same variant with linearly increasing difficulty, but use a different player skill for each line. The black line is the highest skilled player and the lightest blue line is the lowest skilled player. Recall that we increase simulated player skill by decreasing the standard deviation of the time adjustment.

The resulting empirical probability distributions from the experiment, shown in Fig. 5a, are especially interesting. We can see that players of low skill experience a very different game from high skilled players. The low skilled player finds that a score of 0 is most likely, meaning they don't experience any positive feedback early on to encourage them. The high skilled player however has some early notion they are doing well as the most likely score for them is around 25, and it is very unlikely they will achieve a score ≤ 5.

In Fig. 5b, we see that the hazard rates derived from the data are still approximately linear as in Figure 4, but here the intercept a is changing as well as the slope b. Because hazard rates can not be negative, the black hazard line flattens near the origin, while the trend of the line is towards a negative y-intercept a. Easier parts of the game are trivial and unlikely to lead to the AI failing, which causes a flat hazard rate. Its not until the line starts turning upward that this AI begins to experience a challenge.

It is important to reiterate that the shape of the probability distribution is dependent on the player's skill – the low skilled player and high skilled player do not experience the game in the same way. This quantitatively impacts the designer's ability to make a game that can please all players without making some sacrifices on game balance.

5 Modeling Learning

So far we have only looked at examples where the hazard rate derived from the experimental data is increasing, as we tend to find that games become more difficult as a player progresses. We now model players improving over time as they repeatedly play the same game. Learning is typically modeled with a power law function:

$$T = A + B(n + E)^{-R} \tag{7}$$

where the time T to perform a task decreases as the number of repetitions n increases, A defines the best possible time to achieve the task, B defines the performance on the first trial, $R > 0$ is the learning rate, and E represents prior experience (Lane 1987). Power law functions model improvement which goes quickly at the beginning, but then slows down as the player learns the easiest ways to improve, but then takes more time to develop the ability to improve at higher level skills.

Instead of modeling a decrease in time to complete a task, we model a decrease in player error, which has similarly been shown to follow power laws. In our system, we model this improvement by decreasing the standard deviation of the time adjustment after each play. To generate the data, we simulate 50,000 AI players, each repeating the game 10 times. After each of the 10 times, the standard deviation is reduced to follow the power law learning equation. We vary the learning rate R for each test to explore the learning effect.

5.1 Generalized Pareto / Hyperbolic Hazard

We can see the empirical results of modeling learning in Figure 6. This is the same game as in Sec. 4.1 with constant belt speed, but now each player has a learning rate R. Black has no learning, and light green has fastest learning. From the empirical distributions in Fig. 6a we can't tell exactly what the distribution may be, but the log plot in Fig. 6b shows that faster learning rates cause a larger departure from exponential (as exponential curves are lines in a log plot).

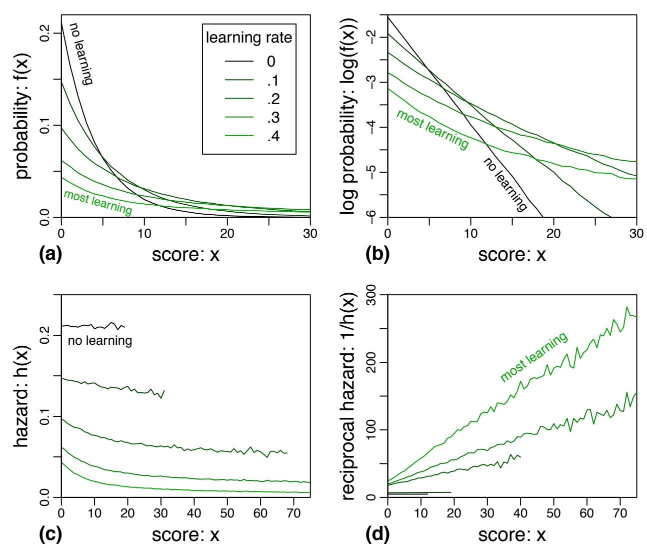

Figure 6: Data collected from simulated games with constant belt speed when modeling learning effects. (a) Empirical PDF follows the Generalized Pareto Distribution. (b) Log plots show faster learning increasingly diverges from exponential. (c) Hazard rates shows a decreasing trend as the learning rate increases. (d) Reciprocal hazards $1/h(x)$ derived from the data are linear, which fit the Generalized Pareto Distribution.

By deriving the hazard from our data, as shown in Fig. 6c, we see for the first time a decreasing hazard rate with increasing score x. The decreasing behavior arises because with repeated plays, the player is learning and improving, which makes higher scores easier to obtain. The hazard decreases faster with a higher learning rate, and reduces to a constant hazard for a zero learning rate. We can see in Fig. 6d further evidence these curves are hyperbolic (i.e. reciprocal linear) hazard rates, when free parameters for inverting the hazard are set appropriately.

We theoretically model this using hyperbolic hazards:

$$\begin{aligned} h(x) &= & & a + \frac{b}{x+c} \\ H(x) &= \int_0^x h(u)du = & & ax + b\log(1 + x/c) \\ S(x) &= e^{-H(x)} = & & e^{-ax}(1 + x/c)^{-b} \\ f(x) &= h(x)S(x) = & & \left(a + \frac{b}{x+c}\right) e^{-ax}(1 + x/c)^{-b} \end{aligned}$$

where a is related to initial difficulty at the start of the game, b determines the learning rate, and c allows us to have scores $x = 0$ and helps model previous experience. These equations model the Generalized Pareto Distribution (Leemis and McQueston 2008; Leemis et al. 2012), commonly used to understand extreme events such as floods and earthquakes.

6 Analyzing Actual Game Distributions

We now apply this type of survival analysis to examine the distribution of scores from flappybird.io (McDonnell 2014), a popular web-based version of the original Flappy Bird. As explained in Sec. 4.1, Flappy Bird has a constant difficulty, so without learning effects would exhibit a constant hazard rate and an exponential probability distribution.

Figure 7 shows the actual score distributions for 4 months from March 2014, when flappybird.io first launched, to June 2014. This time period covers over 175 million individual plays. The spike at the left of the graphs occur because the first pipe is easier to score in Flappy Bird due to scoring at the center of pipes and setup time for the first pipe. It is not apparent from Fig. 7a which distribution is occuring, but non-linear log plots in Fig. 7b shows it is not exponential.

By deriving the hazard from the data, we see in Fig. 7c that the hazard rate decreases rapidly, indicating that learning and past experience may be a factor at making the game less difficult for higher scoring players. Plotting score vs reciprocal hazard rate $1/h(x)$ in Fig. 7d shows a linear relationship, indicative of the Generalized Pareto Distribution. This shows the same distribution as we showed with simulation in Sec. 5.1, giving evidence we are dealing with learning effects.

The hazard rate curves appear to be proportional hazards (Kleinbaum and Klein 1996), but show a trend in Fig. 7c where later months have lower hazard rates at higher score values. We hypothesize this is due to (1) players having more time to practice and improve and (2) poorly performing players becoming frustrated and exiting the sampling pool, which shifts the probabilities towards more skilled players. Although the difference in the graphs appear slight, because of the high number of samples these effects are significant.

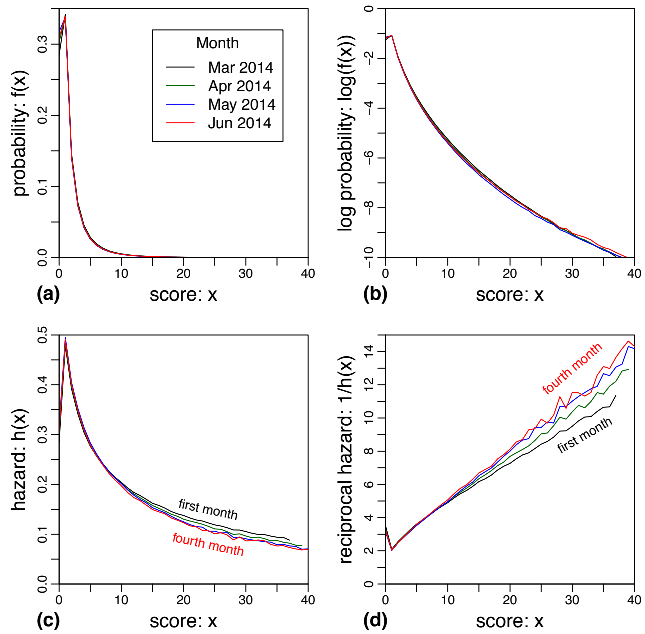

Figure 7: Actual, not simulated, player data from over 175 million plays of flappybird.io matches a Generalized Pareto distribution. (a) The probability spike occurs because in Flappy Bird the first pipe is easier to pass than the rest. (b) Non-linear log probability means the distribution is not exponential (c) Hazard rates show a divergence at higher scores. (d) Divergence is more apparent with the reciprocal hazard.

7 Conclusion

Although our simulated game was invented for simple experimentation, the computer-aided game design and survival analysis techniques presented here can be used to improve more complicated games during the design stage. Even though we have shown it is statistically challenging to create a game variant that is well balanced for all skill levels, examining hazard rates can help tune a game for a specific range of player skills, giving insight into how difficulty changes as a player repeatedly plays a game. We have used similar analysis to show power law hazards lead to Weibull distributions, and exponential hazard rates lead to the Gompertz distribution, but did not have room to present this here. Finally, we have presented evidence that learning rates should not be ignored when modeling players, and we hypothesize for future work that one can find an optimal rate to increase difficulty to keep up with the loss of difficulty due to the natural learning rate. This could lead to an effective constant difficulty to players, which would, in theory at least, keep players in flow.

8 Acknowledgments

Thank you to James Marion, creator of Peter Panic, for inspiring the box packing game, to Adam M. Smith, Julian Togelius, and Dan Gopstein for helpful discussions about the paper, to Max McDonnell for providing the analytics for flappybird.io, and to Dong Nguyen for creating the original Flappy Bird.

References

Bauckhage, C.; Kersting, K.; Sifa, R.; Thurau, C.; Drachen, A.; and Canossa, A. 2012. How players lose interest in playing a game: An empirical study based on distributions of total playing times. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, 139–146. IEEE.

Boyce, W. E.; DiPrima, R. C.; and Haines, C. W. 1992. *Elementary differential equations and boundary value problems*, volume 9. Wiley New York.

Cleves, M. 2008. *An introduction to survival analysis using Stata*. Stata Press.

El-Nasr, M. S.; Drachen, A.; and Canossa, A. 2013. *Game analytics: Maximizing the value of player data*. Springer.

Feng, W.-c.; Chang, F.; Feng, W.-c.; and Walpole, J. 2005. A traffic characterization of popular on-line games. *Networking, IEEE/ACM Transactions on* 13(3):488–500.

Isaksen, A.; Gopstein, D.; and Nealen, A. 2015. Exploring game space using survival analysis. In *Foundations of Digital Games*.

Kleinbaum, D. G., and Klein, M. 1996. *Survival analysis*. Springer.

Lane, N. E. 1987. *Skill acquisition rates and patterns*. Springer Science & Business Media.

Lee, E. T., and Wang, J. W. 2013. *Statistical methods for survival data analysis*. John Wiley & Sons.

Leemis, L. M., and McQueston, J. T. 2008. Univariate distribution relationships. *The American Statistician* 62(1):45–53.

Leemis, L. M.; Luckett, D. J.; Powell, A. G.; and Vermeer, P. E. 2012. Univariate probability relationships. *Journal of Statistics Education* 20(3).

McDonnell, M. 2014. flappybird.io. `http://flappybird.io`.

Muller, H.-G., and Wang, J.-L. 1994. Hazard rate estimation under random censoring with varying kernels and bandwidths. *Biometrics* 61–76.

Nealen, A.; Saltsman, A.; and Boxerman, E. 2011. Towards minimalist game design. In *FDG*, 38–45. ACM.

Nelson, M. J. 2011. Game metrics without players: Strategies for understanding game artifacts. In *Artificial Intelligence in the Game Design Process*.

Nguyen, D. 2013. Flappy bird. Apple App Store.

Rinne, H. 2014. *The Hazard rate : Theory and inference (with supplementary MATLAB-Programs)*. Justus-Liebig-Universitt.

Rosenberg, P. S. 1995. Hazard function estimation using b-splines. *Biometrics* 874–887.

Swink, S. 2009. *Game Feel*. Morgan Kaufmann.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games*.