PREDICTING VIDEO GAME SALES USING AN ANALYSIS OF INTERNET MESSAGE BOARD DISCUSSIONS

A Thesis

Presented to the

Faculty of

San Diego State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

In

Linguistics

---

by

Steven Emil Ehrenfeld

Spring 2011

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Steven Emil Ehrenfeld:

Predicting Video Game Sales Using an Analysis of Internet Message Board Discussions

Robert P. Malouf, Chair
Department of Linguistics and Asian/Middle Eastern Languages

Eniko Csomay
Department of Linguistics and Asian/Middle Eastern Languages

André Skupin
Department of Geography

April 19, 2011
Approval Date

ABSTRACT OF THE THESIS

Predicting Video Game Sales Using an Analysis of Internet
Message Board Discussions
by
Steven Emil Ehrenfeld
Master of Arts in Linguistics
San Diego State University, 2011

Analysis of language posted publicly on the internet has given us a new way of observing consumers, and natural language processing methods allow us to analyze this large amount of text as it is being posted online. In this study we build and analyze corpora of internet message board discussions and use this analysis to build a model that attempts to predict videogame sales figures. Weekly corpora are built by downloading and processing text consisting of the discussions of a large community focused on the topic of videogames. This text is then analyzed to determine which videogame titles generate the most discussion within the community for each week. We use support vector regression to create a model that is able to make predictions about future sales figures. Similar methods have been successful in predicting success in other areas such as box office ticket sales for movies and book sales. By tracking and analyzing consumer interest within online communities, these methods can provide a number of industries with the ability to predict the success of products.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

CHAPTER 1

INTRODUCTION

The enormous amount of text posted publicly by Internet users each day makes it possible to observe the public discussions of a variety of communities, and the flexibility and accessibility of this Internet data makes it an attractive source of information about consumers. Because the amount of language data posted daily within an Internet community is too large to be analyzed by human readers, automated methods are needed in order to make practical use of this type of data. Natural language processing can be used for a variety of tasks, one of which is to analyze large corpora of text in order to learn something about the language being used, and this analysis can be used to make predictions about future events such as consumer purchases. In this study we build a large corpus of text and analyze it to determine what a community of videogame fans are talking about. We then use statistical methods to create a model that compares the data obtained from discussions to sales data, and attempts to predict which games the community will buy.

STATEMENT OF THE PROBLEM

The videogame industry has shown a great need for accurate sales data. This study aims to investigate how an analysis of Internet discussion can help predict the buying habits of a community of videogame fans. We hypothesize that the games that are selling the most are also the games that people are talking about most, and that we can use this assumption to predict videogame sales before sales data has been released. The study involves gathering a large amount of Internet text in a corpus, analyzing it to determine which games are being

discussed the most, and using this analysis to create a model that is able to predict how well a game is selling. If a model created from corpus data gathered from Internet discussion boards can be used to predict the sales performance of videogames, such a model could prove useful to a number of industries that are interested in forecasting sales data.

PURPOSE OF THE STUDY

As the videogame industry has grown since its inception in the 1970's there has been an increase in the demand for software sales charts, data that is important to the videogame industry, its investors, videogame journalists, and the online community of videogame fans. Like the film, music, and publishing industries, the videogame industry uses sales data when making a variety of decisions. For example, a game company will want to keep track of the sales of its competitors' products and investors will want to keep track of sales when planning investments. A game publisher's marketing department may use sales data to track the effects of an advertising campaign, or to study which demographics a game is popular with so that it can use its budget effectively. Additionally if a game is a surprise success in its first weeks, its publisher may use this information to decide whether or not to order more copies from the manufacturer in order to meet consumer demand. An increase in the availability of sales data can be used by the industry when making these kinds of decisions. The gaming community is mostly interested in sales data as entertainment. Members discuss sales data online and sometimes attempt to predict sales data before it has been released. Fans often become loyal to certain hardware manufacturers and watch sales data hoping for the success of their favorite hardware or software, in a way similar to how sport fans follow the statistics of their favorite teams. The website *The simExchange* hosts a stock market where users can invest a virtual currency in the games of their choosing and compare the accuracy

of their predictions with other members of the community (http://www.simexchange.com/). If an automated method of generating sales data could be shown to be accurate it could be used not only to predict sales data before its released, but also to provide previously unavailable sales data to the videogame industry. The videogame industry would be able to study a specific community and determine which games it is interested in. Because there are communities focused on many different demographics, a business could look at an online community of female gamers, older gamers, or casual gamers to determine which types of games each demographic shows the most interest in. If a game is found to be popular with a community of female gamers it may then choose to use this information to place a commercial during a television show popular with females. This method could also be used to track interest in games on a daily basis, providing information that weekly or monthly sales chart are not able to provide. If found to be practical and accurate enough for real world use, an automated method of tracking how games are selling based on corpus data could provide a new way of determining the popularity of games across Internet communities.

LIMITATIONS OF THE STUDY

This study is limited by the availability of both sales data and discussion data. There are two main sources of videogame sales data available to the general public on the Internet. First are the charts released by the NPD Group, a market research company that provides sales data to the videogame industry. Though a limited set of this data is made available to members of the Internet community for free, the NPD Group's main market is the videogame industry, whose members pay the company for use of its data. The other source is the website VGChartz (http://www.vgchartz.com/) which provides similar data on the Internet for free and is used mainly for entertainment purposes by the gaming community. While there is

debate on the relative accuracy of the NPD numbers and the VGChartz numbers (Passarella, 2008) for the purposes of this study the data provided by VGChartz is most suitable. Most importantly, VGChartz has allowed us free use of its data, while NPD Group has stated that it does not respond to requests from students (The NPD Group, n.d.). A business with paid access to NPD data may be able to use NPD numbers or a combination of sources to produce more accurate predictions. Additionally while the NPD Group releases its data as a monthly top ten list, VGChartz releases a list of the top 30 games weekly, giving us more data to work with. Both the NPD Group data and the VGChartz data are estimations created using a variety of methods such as online sales charts like those on amazon.com, polls and surveys, and information provide by retailers (Javal, 2010; VGChartz, n.d.). Though neither set of numbers is perfect, they are the sources of sales data that the industry and the Internet community rely on. Because we can only test our predictions against the best estimations that we have available and not exact sales numbers we will not know how close our predictions come to real world sales figures, only how similar they are to the best available estimations. For the purposes of this study we are making the assumption that the available sales data is accurate.

The discussion data that has been compiled for use in the corpora also has limitations. While a perfect corpus would include a representative sample of all videogame buyers, this is not possible using only text posted publicly on the Internet, as we know very little about the users who are posting the text online. There are a variety of communities where gamers can discuss their hobby each populated by a different demographic, with some catering to a specific market such as female gamers or gamers over a certain age. Information about individual users is rarely provided unless a thread or poll is created specifically for the

purposes of posting personal information. The discussion data that we have available for this study has been posted by a self selected community of game fans, and can not be assumed to represent all game buyers. Further study would be needed in order to determine how well each community represents the market as a whole. Instead of attempting to create a corpus representative of the entire gaming community we instead focus on one large community of gamers. The data obtained from this corpus will therefore represent this single community and not the population as a whole.

Another limitation is the method used to determine which games community users are talking about. Ideally, the method should find all posts in which a user is discussing a particular game and ignore all other posts. Our method looks through the corpora and finds all mentions of a game's name. While this method is relatively simple and does correctly locate most mentions of a game, there are a number of ways in which it could be improved. First, a post may include a discussion of a game but not mention the game by name. A thread about a popular game may have tens of thousands of posts and it is unlikely that users will be typing the topic game's name in each post. For example, these three posts were both found in a thread about the game *Mass Effect 2*:

> 1. Very glad of the changes made to the game. Can't wait to play this Tuesday night. (striKeVillain!, 2010, para. 1)
>
> 2. Loving this game WOWOWOWOW (neojubei, 2010, para. 1)
>
> 3. I saw today that a local Wal-Mart is still selling a copy of the Xbox 360 collector's edition of this game, albeit still priced at $70. Is that worth it for the collector's edition? I looked around, and it seems like it has held its value pretty well. I just bought a 360 earlier this month and started playing the first *Mass Effect*, been looking into my options for buying the second and ran into this. (Baron, 2010, para. 1)

The first post is from a user who will likely purchase the game once it is released, while the second is from a user that is already playing the game after its release. Neither of these posts

mentions the name of the game that is being discussed and therefore will not be counted by a search of the corpus. The third post is from a user that is playing the first game in the series and considering purchasing the second. This post refers to the game series by name and so this post will be counted whether or not the user ends up purchasing the game. This method of search isn't able to distinguish between users who are going to purchase a game and users who are simply discussing the title, and this analysis of the corpus will therefore give us a general idea of the amount of discussion about a game rather than a count of all users that have purchased or plan on purchasing the title. Additionally, a single name may refer to multiple games. For example, the *Call of Duty* series contains multiple games such as *Call of Duty: Modern Warfare* and *Call of Duty: World at War*, and users do not always specify which game they are referring to, often using the abbreviation *COD* to refer to any one of these games or to the series itself. These users are most likely referring to the most recently released game in the series, and a human reader will most likely be able to determine which game the poster is talking about from context though a computer search would have difficulty with this task.

CHAPTER 2

REVIEW OF THE LITERATURE

A number of researchers are using statistical methods in the prediction of sales data in other industries. It is unclear which methods private analysts such as the NPD Group use in generating their sales data and it is possible that online discussions are analyzed in the creation of this data. Like the videogame industry, the film industry has shown a need for sales data and researchers have had success in creating models that use the Internet as a source of data for prediction of success. These studies generally attempt to predict opening weekend sales numbers by measuring the amount of consumer interest and analyzing the sentiment of Internet users.

Predicting Movie Success and Academy Awards Through Sentiment and Social Network Analysis (Krauss, Nann, Simon, Fischbach, & Gloor, 2008) attempts to use the "wisdom of crowds" (p. 1) to predict both Academy Awards nominations and box office revenue by looking at an "expert community"(p. 10). This study looks at the message boards of the Internet Movie Database and analyzes both the quantity and sentiment of discussion about a movie. It also identifies trendsetter users who are especially influential because of their status within the community, and because films released later in the year have been shown to have a better chance of winning an Academy Award, it takes into account how much time has passed since the movie's release.

Predicting the future with social media (Sitaram, & Huberman, 2010) is a study that attempts to predict movie ticket sales by looking at data from Twitter. The data is obtained

from Twitter in the time prior to a movie's release date and analysis of this data is used to predict opening weekend movie revenues. They hypothesize that "movies that are well talked about will be well-watched" (p. 1). The study finds that a model based on the rate at which users are tweeting about a movie can perform better than the current "gold standard" (p. 5), a simulated stock market in which users predict the success of movies. The sentiment of the community is also analyzed in order to determine how positive or negative "buzz" (p. 8) relates to movie ticket sales. For opening weekend sales predictions, the quantity of tweets about a film was found to be more useful than the content of the tweets, with analysis of the sentiment of the tweeting community improving predictions for a movie only after the movie had been released.

Exploring the Value of Online Product Ratings in Revenue Forecasting: The Case of Motion Pictures (Dellarocas, Zhang, & Awad, 2007) attempts to measure consumer word-of mouth about movie releases by analyzing online product review websites. The study then combines this analysis with a number of more traditional metrics used for predicting box office success such as marketing budget and critical ratings and compares each metric's value in predicting a movie's financial success.

The Predictive Power of Online Chatter (Gruhl, Guha, Kumar, Novak, & Tomkins, 2005) focuses instead on the publishing industry and looks at the correlation between blog postings about a book and the book's ranking on sales charts. The study aims to create a system that is able to predict a spike in a book's sales before it occurs. The researchers measure blog activity by automatically generating search queries that are able to determine which books bloggers are talking about. The researchers identify a recognizable pattern of

blog posts that occur before a spike in book sales, and are able to predict these spikes in sales.

Similar methods have also been used in attempts at predicting the performance of stocks. *Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards* (Antweiler, & Frank, 2004) looks at message boards where communities of users discuss stocks and determines that there is "financially relevant" (p. 34) information to be found on these message boards. The discussions have an "economically small" (p. 3) but significant impact on the market, and can be used to predict trading volume and volatility. *Twitter Mood Predicts the Stock Market* (Bollen, Mao, & Zeng, 2011) analyzes the mood of the entire Twitter community and finds that the moods "Calm" and "Happiness" as measured by the Google-Profile of Mood States are useful in the prediction of the Dow Jones Industrial Average.

CHAPTER 3

METHODOLOGY

In order to investigate the relationship between the content of Internet discussions and videogame sales I compiled corpora consisting of text gathered from an Internet message board focused on the topic of videogames. The goal of this investigation is to use corpus data to predict what titles people are buying, with the assumption that people are both more likely to talk about games that they have purchased or are going to purchase and are more likely to purchase games that they see other gamers discussing.

DESIGN OF THE INVESTIGATION

The study has three main parts: First we gather data from an online community, we then format and analyze the data, and finally we use the analysis to create a model that is used to predict the performance of individual games. A description of the programs written for these tasks can be found in the Appendix. In the first step we observe the discussions of an Internet community over a period of time in order to create a number of corpora containing the content of the discussions that are taking place during this time. A computer program is written that is able to automatically download and compile the text of discussions from a specified community over a specified period of time. In the second part we use the corpora to determine what games people are talking about. While it would be possible for a human to perform this task, reading each of the thousands of messages that are posted each week on a popular message board in order to understand the intentions of its users, this would be far too time consuming. Instead we create a computer program that uses natural

language processing methods to read and analyze the corpora of text gathered from Internet discussions. This program outputs a list of games that are being talked about the most by the community that is being investigated, along with a number that quantifies how much discussion there is about each game. There are a number of methods that could be used to measure how much a game is being discussed, and for this study we use a method that simply searches for the name of a videogame and counts how often the names appears in the corpora.

In the third part of the study we use the analysis produced from the corpora, along with past sales charts and game release dates obtained from the Internet, to create a model that predicts how well each game is likely to be selling. To do this, an entry containing the available data for each game is compiled into a format that can be read by the statistical software that we are using. The software then uses this data to create a model using support vector regression to predict sales data. The software outputs predictions as well as a measure of the error rate of the predictions.

POPULATION

I chose a message board called NeoGAF, one of the largest videogame discussion boards on the Internet, as the source of discussion data. One reason that I chose this board is that unlike most, it is not divided up into separate sections for discussions of different kinds of games based on categories such as genre or hardware platform. Instead there is one large forum for all videogame discussion as well as one for off-topic discussions, with gaming discussion accounting for about 67% of posts. This allows for a single corpus consisting of discussions of all types of games. NeoGAF has over sixty thousand registered members and over twenty million current posts as well as a large archive of past postings (Big Boards,

n.d.). Though there are no reliable statistics available on the demographics of the forum's members, it is seen by the gaming community and as a relatively mature crowd of "core" gamers. Core gamers invest more time and money in their hobby than so called "casual gamers", who may only play games occasionally and may play games on their cell phones or handheld devices rather than purchasing more expensive gaming hardware. While casual gamers are a large part of the videogame buying community, these gamers are less likely to spend time online discussing games, though there are communities of more casual gamers available for study as well. Like the gaming community as a whole, the majority of members are assumed to be Male, with an average age of 34 (Entertainment Software Association, 2010), though there are many members outside of this demographic. Unlike most videogame communities, NeoGAF members are generally interested not only in playing games but also in the politics and trends of the videogame industry itself, as well as discussions of videogame hardware and software sales charts. Users post under anonymous names, and like most message boards the number of users who actively post is much smaller than the total number of users who read the forums. As a result the community consists of a small number of frequent posters who are well known to the community, a large number of users who only post once in a while, and presumably an even larger number of users who do not post at all and may not be registered members of the forum (Nielsen, 2006). Topics of discussion are posted as threads in which any user may post his or her own text, though only users who have been a part of the community for a period of time are given the ability to create their own topics. Text that is posted is public and viewable to anyone who visits the site and topics are organized so that the most recent postings appear first. The forum has a strict set of rules and is more heavily moderated than most similar communities. Though much of the

discussion is focused on games that people are planning to buy or are currently playing, there is also discussion of topics such as classic games, announcements of new games, and speculation and discussion of future videogame hardware and technology. Threads are also created for speculation and discussion of sales data that is publicly released every week or month.

TREATMENT

Our data consists of a corpus for each week from February 3rd through July 31st 2008 and February 1st through May 30th 2009. Each corpus is made up of one tenth of all text posted in a particular week in both the Gaming Discussion and the Off-Topic Discussion boards. Because of the large amount of text posted each week, only every tenth post is included, giving about 300,000 words or about 1.5 megabytes of data per week for each of the 43 weeks. A Python program was written to gather this data. Because posts are numbered sequentially by their time of posting, we are able to download all posts from a given week by specifying a range of URLs containing these posts and downloading the data that these URLs point to. This allows the program to go through all of the posts, download the data for every tenth post as an HTML file, and compile these files into a separate corpus for each week. This gives us files containing all posted material as well as all HTML code for a week of discussion. The program then extracts only the user-posted text, which is identifiable by HTML tags, removes all punctuation, and converts all letters to lowercase. We are then left with only the language content posted by users, discarding all HTML code and other meta-data. The program does not remove text in which a user quoted another user's post as is common on message boards. We also create files containing the sales charts for each week and a list of games released in each week, compiled from data available on the Internet.

When making predictions, the model that we create will not have access to the current week's sales charts, but will use information from sales charts from the previous two weeks in it's predictions. In the next step the discussion corpora will be analyzed and combined with this sales information to give us an entry containing all available data for each game.

Data Analysis Procedures

I next wrote a program that analyzes the discussion corpora in order to determine which games are being talked about the most. This python program outputs a number for each game that tells us how popular a game is with the community in the current week, as well as a number for each of the previous two weeks. While a computer readable corpus makes this basic task of reading through the text and counting the number of references to a game simple, there are a number of factors that complicate the process. First, it is not obvious which games will appear on the sales charts for any given week until the sales data for that week is released. In a real world setting, this information would not be available until sales charts are released at the end of the week, at which point a prediction of sales figures would be unnecessary. Therefore we must predict not only the sales figures themselves, but also which games will appear on the sales charts. In order to make a guess about which games are worth looking for in the text, we looked at past sales data and determined that of the many games that are discussed each week, with few exceptions the games that appear on sales charts are either new releases or games that also sold well in the previous week. The program therefore takes in a list of the current week's new releases and a list of the games that appeared on the previous week's sales chart and combines them into a list of games that should be searched for and counted in the current week.

Another issue is that in the casual language of an Internet message board, a game is generally not referred to by its official title and therefore a simple search will miss many references. For example the game with the name *Grand Theft Auto IV* appears at the top of the sales charts for much of 2008, but message board users refer to this game by a number of names and abbreviations including *GTA4*, *GTAIV*, and *Grand Theft Auto 4*. The program has to be able to search for all of these possible names but understand that they refer to a single game title. The program uses regular expressions to search the corpus files for a number of names for each game. Regular expressions are lines of text that can be used to search through text files. The program uses regular expressions to search the corpus and returns each line that contains a match for one or more of the possible names that the regular expression specifies. A perfect regular expression would be able to give us every post in which a user refers to a game, though in practice generating a regular expression that cover every possible name for a game is a difficult task.

An example of a regular expression for *Grand Theft Auto 4* is: (gta4|grand theft auto 4|grand theft auto iv|gtaiv|gta four|gta iv|gta 4). When a match is found in a file for any of these names, the program records a match for the first entry in the list. The first entry in this regular expression is the name that the program uses to keep track of each game, and it is followed by a number of possible alternative names. Both hand written and automated methods of generating regular expressions from game titles were tested, each introducing a number of advantages and disadvantages.

It is likely that due to human error there are some names that appear in the corpus that are missing from the large list of hand written regular expressions, and this surely creates some error. On the other hand, a human has the advantage of being familiar with the types of

abbreviations that are and aren't used in the real world and can use this intuition when compiling regular expressions.

The automated method reads in the game's official name and attempts to create a regular expression that only matches the names that Internet users would be likely to use to refer to the game. This generates a large list of names for each game, some of which are unlikely to be used. Though the computer automated method speeds up the process considerably and eliminates the human bias and error involved in writing a large number of regular expressions, it does introduce some errors of its own. A number of possible names must to be left out to avoid false matches and both a computer program and a human have to be able to identify these names and make a decision to either include a name or not. If an abbreviation matches the spelling of a common English word it must be removed to avoid counting uses of this word that do not refer to the game's title. One example is the game *Star Ocean*, which could be abbreviated as *so*. This name had to be removed to avoid matches with the common English word "so", and message board users would likely avoid this abbreviation for the same reason. If this name had been included, the program would have erroneously predicted *Star Ocean* to appear at the top of the sales charts every week. This name was removed and the regular expression for this game may miss some posts for this reason. On the other hand, the *Call of Duty* series is commonly abbreviated as *cod* which does match an English word, though we assume that users are referring to the series of games, as the English word "cod" is not often used in this context. Game names are most often abbreviated by using the initials of each word, including non-capitalized words. *The House of the Dead* series for example is abbreviated HotD, while the most common method of abbreviation in English would leave out the un-capitalized words giving us HD, an

unlikely abbreviation for this series. Both a human and a computer program must be able to either choose the most likely style of abbreviation or include a number of possible abbreviations. The automated method has difficulty with games that have non-standard abbreviations such as *Killzone 2* (kz2) and *Mario Kart DS* (mkds). If a game contains a number, the name will be included with both modern and roman numerals as well as without numbers. If a game includes a subtitle, such as *Fire Emblem: Shadow Dragon*, the name will be included with and without the subtitle, and the subtitle itself will be included as a name as well. The automated method is not able to selectively choose which words to include and which to leave out like a human writer is. For example, a human might decide that the regular expression for the game *New Play Control! Mario Power Tennis* should include the name "*Mario Tennis*" as this a natural name to use for the game, while the automated method is unable to make this decision.

Table 1 shows the results for both of these methods measured in precision and recall. Recall tells us how many of the total references to a game in the text are found by a regular expression, while precision tells us how many of the matches that we find are referring to the correct game. In this test the hand written method outperform the automated method of generating regular expressions in both precision and recall. The precision of both methods is lower since regular expressions will often match references to other games within the same series with similar names, though these matches do give us information about the amount of interest in the series. A more advanced program could possibly be written that would be able to correctly produce the names that users would and would not use to refer to a game, though at this time, writing out regular expressions by hand is the simplest and most accurate method.

Table 1. Accuracy of Two Methods for Generating Regular Expressions

	Precision	Recall
Hand-written regular expressions	88.28%	92.95%
Automated regular expressions	79.65%	82.82%

Once the program has read in these regular expressions from a file, it uses them to count the number of lines that contain one or more mentions of each game. In the program this number is known as $_{w0}$Count. Each time that a match for one of these names is found in the corpus for the current week, the $_{w0}$Count for the game is incremented. The same counts are then compiled for the previous two weeks ($_{w-1}$Count and $_{w-2}$Count). These counts are combined into a entry, along with the corresponding sales data for each week ($_{w-0}$Sales) and the previous two weeks ($_{w-1}$Sales and $_{w-2}$Sales), as well as a number representing the inverse of the number of weeks since the game has been released (1/Age). This gives a list of entries for each game with the format: [Name, $_{w-2}$Count, $_{w-1}$Count, $_{w0}$Count, Age, $_{w-2}$Sales, $_{w-1}$Sales, $_{w-0}$Sales,].

The list is then formatted into a comma-separated .arff file for use with machine learning software. Weka (Hall, Frank, Holmes, Pfahringer, Reutemann, & Witten, 2009) is free open source software that provides a number of machine learning methods (http://www.cs.waikato.ac.nz/ml/weka/). These methods can be used to build a model for predicting unknown data based on known data. In this study we use known data gathered from past online discussions as well as data from past videogame sales charts to predict unknown future videogame sales figures. The software reads in the .arff file and uses it to create a model that will use the first six numbers in each line to make a prediction about the last number in each line, allowing us to predict how a game is going to sell. As shown in

Figure 1, the .arff file lists the variable names and types as a header, and then lists the data for each game after the @DATA tag. A number of Weka's prediction methods were tested with varying accuracy and computational efficiency. Once Weka has given its prediction of future sales, it compares the prediction against the actual sales data and calculates the accuracy of the model.

```
@RELATION gamesales
@ATTRIBUTE gamename string
@ATTRIBUTE weekminus2count NUMERIC
@ATTRIBUTE weekminus1count NUMERIC
@ATTRIBUTE week0count NUMERIC
@ATTRIBUTE weekssincerelease NUMERIC
@ATTRIBUTE weekminus2sales NUMERIC
@ATTRIBUTE weekminus1sales NUMERIC
@ATTRIBUTE week0sales NUMERIC

@DATA
linkscrossbow,2,1,1,14,28,23,34
mightandmagicelements,11,9,8,2,?,26,?
dmc4,82,54,64,3,276,148,65
masseffect,590,655,662,14,20,22,?
burnoutparadise,31,17,14,5,28,25,?
theclub,6,2,6,1,?,?,34
madden08,27,36,26,28,17,21,?
brainage2,3,2,1,27,19,30,25
wiiplay,10,5,7,53,75,47,74
turok,11,7,2,4,77,38,34
rockband,18,22,21,14,24,37,45
halo3,17,14,22,22,26,32,39
apollojustice,1,5,1,1,?,?,32
lostodyssey,14,21,24,2,?,83,49
```

Figure 1. Sample of an .arff file for use with Weka

CHAPTER 4

RESULTS AND DISCUSSION

Weka makes available a number of methods that can be used for prediction, each making use of a different statistical procedure and each suitable for use with certain data types. Using these methods we are able to create models that use classification or regression to make predictions.

PRESENTATION OF THE FINDINGS

The method that has been most successful in the prediction of sales data is SMOreg, an implementation of support vector machines for regression. Support vector machines are most often used to categorize data into one of several discrete classes with minimal error, a problem called classification. This method attempts to find a boundary that correctly divides the training data into classes, and then uses this boundary to classify unseen data. In a simple two-dimensional case the boundary will be a straight line, though the input data can be mapped to a higher dimensional feature space using kernel functions if the data is not linearly separable. Rather than using the entire set of training data, support vector machines uses a subset of data points along the boundary, called support vectors. It builds a boundary by finding the function that maximizes the distance between these support vectors and the boundary line, called the optimal separating hyperplane. Unclassified data points can then be classified by calculating which side of the hyperplane each point falls on.

Support vector machines can also be used for regression. Once the data has been mapped to a higher dimensional feature space, linear regression is performed to give us a

function that is used for making predictions. While classification with support vector machines divides the data into two classes and outputs a binary digit based on which side of the hyperplane a piece of input data falls on, support vector regression outputs a real number, making it more appropriate for this type of problem. In order to make a prediction the unknown dependent variable is computed using known independent variables and the hyperplane function. In this case the independent variables are the discussion data, the past sales data, and the age of the game, and the dependent variable is the sales data that we are predicting (Smola, & Schölkopf, 2004).

Results are given as a mean absolute error, the average of the errors for each of the predictions, in units of thousands of copies sold. These results are listed in Table 2. This is calculated by the formula $MAE = \frac{1}{n}\sum_{t=1}^{n}|F_t - A_t|$, where F is the predicted value and A is the actual value. I also calculated the mean absolute percentage error for the predictions, which gives us the error as a percentage of the actual sales numbers, as shown in Table 3. The formula for mean absolute percentage error is $M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$. Using Weka's default options for SMOreg regression, our model gives us a mean absolute error of 25.9622.

Table 2. Results in Mean Absolute Error

	Only New Games	Only Old Games	New + Old Games
Only Sales Data	1657.0476	14.9068	25.9622
Only Discussion Data	91.8538	24.9546	33.0624
Sales + Discussion Data	91.8538	14.8515	25.954

Table 3. Results in Mean Absolute Percentage Error

	Only New Games	Only Old Games	New + Old Games
Only Sales Data	45.07%	36.39%	40.36%
Only Discussion Data	136.09%	42.37%	47.25%
Sales + Discussion Data	136.09%	36.11%	40.60%

The accuracy improves when instead of including the age of a game as a separate variable, the inverse of the age is multiplied by the $_{w0}$Count variable from the discussion data: [Name, $_{w-2}$Count, $_{w-1}$Count, (1/age * $_{w0}$Count), $_{w-2}$Sales, $_{w-1}$Sales, $_{w-0}$Sales,]
This gives us a mean absolute error of 23.7434. These improved results are given in mean absolute error in Table 4 and in mean absolute percentage error in Table 5.

Table 4. Results in Mean Absolute Error with (1/Age * $_{w0}$Count)

	Only New Games	Only Old Games	New + Old Games
Only Sales Data	1657.0476	15.1227	25.9974
Only Discussion Data	91.8538	24.5775	30.8408
Sales + Discussion Data	91.8538	14.8454	23.7434

Table 5. Results in Mean Absolute Percentage Error with (1/Age * $_{w0}$Count)

	Only New Games	Only Old Games	New + Old Games
Only Sales Data	450.72%	34.58%	40.59%
Only Discussion Data	136.09%	41.68%	51.60%
Sales + Discussion Data	136.09%	34.56%	35.45%

We also need to determine whether or not our corpus data is contributing useful information to our model, or if a model that looks at only past sales data would perform as well or better. By removing discussion data and creating a model using only sales data we can determine how much the discussion data is helping the model. When only past sales figures are used, the model predicts with a mean absolute error 25.9974. When discussion data only is used, the model gives us a root mean squared error of 30.8408. This shows us that both the sales data and discussion data sets together are needed to create our most successful model. It is clear that the data gathered from the discussion corpora does is in fact create a more useful model for predicting sales data than looking at past sales alone.

When new releases are removed from the data (leaving only games with an age greater than one week), the accuracy of the model improves greatly, giving us an error of 15.1227. When new games (with an age of one week) are considered on their own we get a mean absolute error of 91.8538.

DISCUSSIONS OF THE FINDINGS

From the results in Table 4 we can see that while a model created with sales data alone does a good job of predicting sales figures for games that have previously appeared on sales charts, discussion data is needed to predict sales figures for new releases. This is because new releases have no previous sales data to rely on and therefore analysis of the discussion corpora is the only source of data. As expected, when discussion data is removed it becomes impossible to accurately predict the performance of new releases and the model makes a prediction of 1,779,000 copies for each game, a mean absolute error of 1657.0476. Older games on the other hand are easier to predict, as once a game has been released it follows a more predictable pattern from one week to the next. Though there are some rare

cases in which a game's sales increased after initially weak sales, games generally experience a spike during the week after its release followed by a decline for the next several weeks or months before dropping off the sales charts. When predicting both new and old games together both discussion and sales data are needed to create the most accurate model.

While a better understanding of older games could help in predicting the rare occasions in which a game's sales increases over time after its release, past sales data is enough to create a model that can predict the sales of old games with some accuracy. Sales predictions are most useful to the videogame industry in the time before a game's release, and it is in the prediction of first week sales numbers that information gained from the analysis Internet discussion data has shown to have the most value. There are a number of reasons why it is easier to predict sales figures for old games than for new releases. First, new releases tend to either appear at the top of the charts with very high sales numbers, or not appear on the charts at all, and this variability makes them difficult to forecast. For example, the game *Resident Evil 5* sold 1,011,000 copies in its first month month. Due to the large amount of discussion for this game the model correctly identified it as a big seller with a prediction of 430,315, though this prediction has an absolute error of 580,685. One reason for this difficulty in predicting new releases is that while the data for each week contains twenty to thirty older games, of the many games released each week only three to four new releases make it onto the sales charts. This gives us less training data for the model to use in predicting sales for new releases. New releases generally fall into one of three categories. Some games are considered "shovelware" by the online gaming community and are ignored almost completely. These games are generally made with low budgets to maximize profits and appeal to uninformed buyers such as parents looking to purchase a game for their

children. Most games that are based on popular licenses from television programs or films are considered to be shovelware. These games sometimes sell very well, but exist in a market outside of the community investigated in this study, so while they may have sales data after they are released, they rarely have any discussion data from the corpus. Additionally many games popular with casual gamers do not appear on the available sales charts because they are played on computers or mobile phones rather than dedicated videogame hardware. Another category is niche games. These games are often produced by small studios and are made specifically to appeal to the population of core gamers. These games find a small but devoted audience. They are generally more complex than casual games, are often imported from Japan, and do not appeal to the mainstream audience. These games do not sell well and generally do not appear on sales charts, though they are a popular topic of Internet discussion and are therefore overrepresented in the corpus. The third type of new release is the "AAA" title, a game made by a major game studio with a large development and advertising budget. These games are popular among a wide audience of gamers and appear at the top of sales charts for their first few weeks, generally staying within the top 10 for months. As well as having the largest sales numbers they are the most talked about, with online discussions sometimes beginning months or years before their release. Unfortunately the model does not have much data on these games, as there are few games of this type. The difficulty in differentiating between these types of games, combined with the limited amount of training data available for new games creates a model that has difficulty predicting how new games will sell.

CHAPTER 5

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

One way that the results could be improved is by compiling a corpus of discussion data from a population that more accurately represents the population described by the available sales data. The population posting on the NeoGAF message boards is a group of gamers that choose to spend their time reading and writing about videogames, and we cannot expect this community to be representative of the game buying population as a whole. The games that become popular with the more mainstream casual gaming market are often big budget titles that sell tens of millions of copies, while core gamers such as those that join internet gaming communities such as the one being studied often prefer niche games that may sell only tens of thousands of copies, not enough to appear in sales charts. Though these core gamers most likely spend the most money on their hobby, there is also a large population of casual gamers that are not being accounted for in the corpora as they either post in different Internet communities or do not discuss their hobby on the Internet at all. Therefore the model created in this study may be more representative of the community from which it was compiled than the market as a whole. A perfect model would use sales data and discussion data collected from the same population, though this data does not currently exist.

A different method of determining which games are being talked about could also be used to create a more accurate model. It is not always obvious which games a user is discussing in a post and it is possible that a more complex analysis that takes into account

factors such as the structure of the community or the topic of the thread that a post is made in could result in a model that better represents the feelings and buying habits of the community.

Though the data created from the model described in this study can't match the accuracy of the currently available sales charts, it can be produced before a game's release, making previously unavailable information available to the videogame industry. Because accurate sales data is important to a number of industries, there is interest in the improved availability of this data and this study has shown the value of data obtained from the analysis of Internet discussion data in the prediction of videogame sales data.

REFERENCES

Antweiler, W., & Frank, M. Z. (2004). Is all that talk just noise? The information content of internet stock message boards. *The Journal of Finance*, *59*(3), 1259-1294.

Baron. (2010, July 21). *Mass effect 2 |ot|* [Online Forum Comment]. Retrieved from http://www.neogaf.com/forum/showpost.php?p=22496682&postcount=17344

Big Boards. (n.d.). *Video games forums and message boards*. Retrieved from http://www.big-boards.com/kw/video-games/

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*(1), 1-8.

Dellarocas, C., Zhang, X., & Awad, N. (2007). Exploring the value of online product reviews in forecasting sales: The case of motion pictures. *Journal of Interactive Marketing*, *21*(4), 23-45.

Entertainment Software Association. (2010, October 3). *Essential facts about the computer and video game industry*. Retrieved from http://www.theesa.com/facts/pdfs/ESA_Essential_Facts_2010.PDF

Gruhl, D., Guha, R., Kumar, R., Novak, J., & Tomkins, A. (2005). The predictive power of online chatter. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (pp. 78 - 87). Chicago, IL: 10.1145/1081870.1081883

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The weka data mining software: An update. *SIGKDD Exploration*, *11*(1), 10-18.

Javal, K. (2010, September 20). *The npd group's software point-of-sale (pos) and consumer methodology*. Retrieved from http://www5.npd.com/tech/pdf/Data_Methodology_for_Software_Services.pdf

Krauss, J., Nann, S., Simon, D,, Fischbach, K., & Gloor, P. (2008). Predicting movie success and academy awards through sentiment and social network analysis. *Proceedings of the European Conference on Information Systems* (pp. 2026-2037). Galway, Ireland.

Neojubei. (2010, January 26). *Mass effect 2 |ot|* [Online Forum Comment]. Retrieved from http://www.neogaf.com/forum/showpost.php?p=19482722&postcount=3666

Nielsen, J. (2006, October 9). *Participation inequality: Encouraging more users to contribute. Alertbox*. Retrieved from http://www.useit.com/alertbox/participation_inequality.html

Passarella, R. (2008, June 17). *Data on the web: Vgchartz vs. npd* [Web log message]. Retrieved from http://radar.oreilly.com/2008/06/data-on-the-web-vgchartz-vs-np.html

Sitaram, A., & Huberman, B. A. (2010). Predicting the future with social media. *Proceedings of the IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology* (pp. 492-499). Toronto, ON: 10.1109/WI-IAT.2010.63

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199-222.

StriKeVillain!. (2010, January 23). *Mass effect 2 |ot|* [Online Forum Comment]. Retrieved from http://www.neogaf.com/forum/showpost.php?p=19433444&postcount=188

The NPD Group. (n.d.). *Academic requests*. Retrieved from http://www.npd.com/corpServlet?nextpage=contact-us-academia_s.html

VGChartz, . (n.d.). *Vgchartz methodology*. Retrieved from http://www.vgchartz.com/methodology.php

APPENDIX

LIST OF PROGRAMS

Program Name	Description of Function
makecorpora.py	**Input:** - A range of URL's containing the posts for a given week (posts are numbered sequentially by their time of posting). **Function:** - Uses urllib's urlopen() function to download every tenth post in a specified range. - Strips away all HTML code, punctuation, and non-post text, leaving only user posted text. **Output:** - A corpus containing the text of every tenth post made in the specified week.
counts.py	**Input:** - Text files containing the corpora for the current week and the previous two weeks. - Text files containing a list of sales data for the current week and the previous two weeks. Lines have the format: game name, weeks since release, sales. - List of the regular expressions for the week's new releases. - List of regular expressions for the games on the previous week's sales charts. **Function:** - Combines the list of new releases and last week's chart into a list

	of games to search for. - Searches the corpora of the current week and the previous two weeks and counts the number of matches for each regular expression in the game list. - Retrieves sales numbers from files. - Normalize sales data for each week. - Compile the game's sales figures, age, and corpora counts into a dictionary entry for each game with the format: [name, count-2, count-1, count0, age , sales-2 , sales-1 , sales0], where: count-2 = corpora counts for two weeks ago. count-1 = corpora counts for previous week. count0 = corpora counts for the current week. age = weeks since the game's release (1 for new releases). sales-2 = sales data for two weeks ago (0 for games newer that two weeks). sales-1 = sales data for previous week (0 for games newer that one week). Sales0 = sales data for the current week (if available). **Output:** **-** A text file with each line containing all the information in a game's dictionary entry.
FormatforWeka.py	**Input:**

	Text file created in counts.py. **Function:** - Create header for Weka. - Multiply count0 by the inverse of age (optional, improves results). - Format data into a comma separated line for each game. **Output:** - An .arff file containing some or all of the available data, to be read by Weka.