Data Mining

Chris Stone

June 7, 2004

Abstract

Statistics is the fuel that drives modern sports. Records are kept that track everything, from the number of pitches thrown by a pitcher to the average speed of a NASCAR driver. Because of this, there are massive amounts of sports data being stored across the world, most of which is accessible through the internet. Although this is good because it allows people to gather the data that they want, a problem arises because of the sheer volume of data out there. Data mining, or "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data", is a very effective way to sort through unwanted statistics in order to find valuable data.

1 Background

The amount of worldwide information being stored in electronic format has grown tremendously in the last twenty years. In fact, the amount of data doubles every twenty months, and the number and storage capacity of databases is growing at an even faster rate. The

main reason for this boom in data storage is the increased availability and decreased cost of computing power.

There is a downfall, however, to massive amounts of information being available. With increased quantities of data, search time also increases. It has become very difficult to find the exact information that one is looking for, since there is often huge amounts of unwanted data that has to be sifted through.

2 Definition

Data mining, also known as Knowledge Discovery in Databases (KDD), is the

> "nontrivial extraction of implicit, previously unknown, and potentially useful information from data. This encompasses a number of different technical approaches, such as clustering, data summarization, learning classification rules, finding dependency networks, analyzing changes, and detecting anomalies."

3 Introduction to KDD

KDD Stages:

1. Select the data to be mined
2. Preprocess the data
3. Transform the data

4. Mine the data, extracting patters and relationships

5. Interpret and assess the discovered structures

3.1 Select the data to be mined

The first three stages take the most time, approximately eighty percent, as is true with most preliminary work. The first stage is to select the data to be mined. This involved finding a source of data that meets several criteria. First, it should have all of the data that you plan on extracting. In my case, I had to find a website that had all of the hockey player statistics that I intended to use in my fantasy hockey league. Second, the data should be displayed in an easy-to-read format. This means that there should be distinct patterns in the data that can be found easily, so that the necessary data can be extracted (this usually means that the data is displayed in a simple table, and not in any sort of graphical way). Once you have found a good source of data, you are ready to move on to step 2.

3.2 Preprocess the data

Preprocessing the data involves discarding large amounts of data that you are certain has no value to you. For example, when I mined hockey statistics from ESPN.com, I only kept the source code to the one page that I was sure had the statistics that I wanted. I then got rid of most of the source code to that page, since most of the code was useless HTML that had no valuable data. Once the huge amounts of information has been cut down to a small section containing useful data, you are ready to continue to step 3.

3.3 Transform the data

Transforming the data is a simple step, provided steps one and two were done carefully. The manageable chunk that was acquired during step two must be broken up into individual lines that will be mined in step four. The imporantce of step one becomes very clear from now on, because if the data was displayed poorly on the source website, it will be very difficult to mine it. The data must have pattern that are present for every individual group of data (in my case, the HTML line for each player is identical, except for the data itself). The pattern must also be distinct, because you dont want it to be present elsewhere in the source code. If it is present in other places, it will be very difficult to distinguish between valuable data and unwanted information. Once you have individual lines, that you are sure contain useful information, you can proceed to step four.

3.4 Mine the data, extracting patterns and relationships

This is the step where the actual data extraction occurs. The program must know the general format in which the data will be presented, so that it knows where the desired data is. For example, you would have to hard code in a certain section of the source page, telling the program where you expect a piece of information to be. The program then goes to that part of the page, and stores whatever data is there. This is why it is important to find a website with intelligent source code, because you want each data line to be uniform (source code that was written by another program usually works well). If each line is uniform, then you can use the same program to extract data for every line (for every hockey player, in my

case). The program then stores the data in some way (I store my NHL data in my mySQL database).

3.5 Interpret and assess the discovered structures

This step varies widely from case to case. It doesn't really involve data mining, it is simply what happens to the data after it is mined. In my case, I use the data in a fantasy hockey game that I am making. I will also write an AI program that will use the mined data to predict which players will perform the best in the future (so that they can add these players to its team).

4 Regular Expressions

In short, a regular expression is a "formula for matching strings that follow some pattern." Everyone has used regular expressions at some point in their life. "ls *.html" lists every file that ends in .html, regardless of what the actual file name is. Also, when using a search engine, regular expressions are sometimes employed. Searching for "comp*" will find web-pages that contain "computer", "compute", "computation", etc. These are both examples of commonly used regular expressions. As with most subjects, the more you learn about regular expressions, the more powerful they become. Here are some basic regular expression metacharacters and a description of what they do:

4.1 Regular Expression Syntax

. Matches any single character. ex: "d.g" would match "dog, "dug", or "dig" but not "drag"

$ Matches the end of a line.

ˆ Matches the beginning of a line.

⋆ Matches any number (including zero) of occurances of the preceeding character. ex: "a*" would match "a", "aaaaa", or "". Another example, ".*" would match any number of any characters

\ Called the quoting chracter, this is used to tell the computer to treat the following character as an ordinary character. ex: "\." matches only an actual period, and not any single character

[ ] Matches any single character that is contained within the brackets. ex: "r[ou]t" matches "rot" and "rob", but not "rat" or "rout".

[0-9] Matches any character in the range of characters. ex: [0-9] would match any number between 0 and 9, inclusive. [A-Z] would match any uppercase letter.

? Matches 0 or 1 (and not more than 1) occurences of the preceding regular expression. ex: [0-9]? matches any single digit or, if no digit is present, matches nothing and the program moves on.

+ Matches 1 or more occurences of the preceding regular expression. This is different than ⋆, which matches 0 or more of the preceeding regular expression

4.2 Application

Here is an example of a line of code that uses regular expressions. It was taken from statparser.php, which is the data mining program that runs every morning, and gathers NHL data.

```
preg_match("/Id=[0-9]*>([^\s]*) (.*)<\/a>, (.*)<\/td><td class=\"verb10\">
([A-Z]*)\s?<\/td><td class=\"verb10\">[0-9]*<\/td>/", $matches[1][$i], $temp);
```

This is one of the applications of regular expressions that appears in my program. It may look confusing, but when it is broken down into its elementary pieces it is actually very simple. I will explain each section to you.

4.2.1 preg_match()

preg_match() is the php function that is used to parse strings. It is given three parameters, the expected string with regular expressions, the string to be parsed (contained in $matches[1][$i]), and an array that will hold the matches ($temp).

4.2.2 Breaking Down the Regular Expressions

$matches[1][$i] holds a single line from ESPN.com's NHL stats page source code, which is an HTML file. Contained inside the HTML are valuable statistics that I want to get out. First I have the escape character, \, which I must have at the beginning and end of the string. "Id=" tells preg_match() to bypass those characters. If any character that is hardcoded in, like "Id=", isn't actually contained in the string, then preg_match() will fail. It is, therefore,

crucial that I find a site with uniform code that doesn't change from day to day (luckily ESPN.com matches this description). Next is [0-9]⋆, which bypasses as many numbers as there are in a row. This is used because the actual number is not known, so it shouldn't be hard coded into the regular expression (if, for example, a 2 was always present at this place, a "2" could be used instead of "[0-9]"). ([ˆ\s]⋆) (.⋆) is used to read in all of the names. \s means the space character, so ˆ\s means "not the space character". This section, therefore, matches every non-space character in a row, followed by a space, and then every character in a row until the next section of code is matched. The parenthesis tell the computer to store the data in $temp[0] and $temp[1] (if there weren't parenthesis, the data would be read in, and then discarded). The rest of the data is acquired in a similar way, following the rules that were previously layed out (\ is used as an escape character before quotes, and regular text is matched exactly). By the end of this line of PHP code, valuable data has been mined off of ESPN.com, and we are well on our way to creating a fantasy hockey game. See, that seemingly impossible line of code wasn't that bad.

4.3 Implementation

Regular expressions were used to perform several tasks in my fantasy hockey program. The code for each task is contained in statparser.php, which is the "behind the scenes" part of my program. Using regular expressions, it gets all of the necessary player data, as well as the necessary goalie data. It then determines the stats that each player accumulated during the previous night's games, and adds them to their corresponding user's running total. It

then, using regular expressions once again, finds and stores that night's schedule (it finds out which teams are playing, where the game is held, and what time it starts for each game). It uses this to "lock in" the players once their game has started, so they can't be dropped or added while playing. The rest of my program, which is contained throughout the HTML files of my fantasy hockey website, uses the data that statparser.php mines each morning. Because of this, the importance of data mining and regular expression to my senior techlab project is quite obvious.

5 The Program

5.1 Introduction to Fantasy Sports

My senior techlab project is not unlike many fantasy sports progams that currently exist on the internet. A fantasy sport is a way to simulate the ownership of a professional sports team while using statistics from actual games. Leagues are created (from about 8-10 teams per league), and a draft is conducted. Each "owner" takes turns drafting a player that they think will do the best in real life. Once each owner's roster is full, the matchups may begin. After every night of games, statistics are gathered about how the players did in their last game. These statistics (which can include goals, assits, plus/minus, shots, penalty minutes, power play points, wins, shutouts, save percentage, and games played in the case of a fantasy hockey leauge) are added to the running total of their owner. These running totals are compared with each other to determine which team wins the fantasy game. Fantasy leagues used to

be run entirely by hand (and sometimes still are), but the internet has drastically increased the number of fantasy sports users. The demand for good, easy, user friendly fantasy sports programs is very high.

5.2 Login Page

The first page that a user encounters is the login page. It asks for the standard username and password in order to login to a fantasy account. When it has the username and password, it accesses my mySQL database and checks to see it the username exists and that the passwords match up. My database stores encrypted passwords, so the entered password has to be encrypted (using md5) in order to check to see if it is valid. If they dont have an account, there is a link to an account creation page that asks for standard information (it also checks to make sure the requested username doesn't already exist, and it asks for the new password to be typed twice). Assuming the password is correct for the given username, the person is now logged in (I used sessions, so they can go from page to page without having to log back in each time).

5.3 Home Screen

Once you are logged in, you are immediatley met with lots of information. There is a toolbar at the top that lists all of the available pages, as well as three other links (two provide the user with lots of information about how they are doing compared with other users, while the other is a link to change their password). The table that is on the home screen is a standings

list. It lists each user in the league, and their win-loss-tie record.

5.4 Running Totals

If the user clicks on one of the two information links not in the toolbar, they can see how they are doing. It shows every user's running total (either season total or weekly total, based on which link they clicked) for each of the ten categories (goals, assists, plus/minus, shots, penalty minutes, and power play points for their skaters, and games played, wins, save percentage, and shutouts for their goalies). This is useful for users to determine where their team is weak, so they can aquire/trade for players that can help strengthen these holes.

5.5 Roster

The roster page is the only page that is unique for each user. It lists exactly what players are on their team, and divides them into three categories: goalies, starting players, and benched players.

5.5.1 Goalies

Each team can have a maximum of two goalies. The roster pages lists these two goalies (or has an empty row if one of the slots is not filled), and their season totals for the four goalie stats. Games played, or GP as it is listed, is the number of hockey games that the goalie has played in during the season. The more games a goalie plays the better, since it means that your goalie is reliable and resistant to injury. The second category is wins (W). You also want this to be high, becuase it means you goalie knows how to win. Save percentage

(SV(this is an accurate measure of how good a goalie is). The last category is shutouts, or SO, which is the number of complete games the goalie has played in where they didn't allow any goals. The final column allows the user to drop one or more of their goalies, thus freeing up a goalie slot. It also lists the NHL team and nightly opponent for each goalie.

5.5.2 Starting Players

There can be a maximum of five starting players (three forwards and two defensemen) that are listed in the "Starting Players" table. Each starting player has their own row, which contains their season stats in each of the six categories (goals, asssits, plus/minus, shots, penalty minutes, and power play points), as well as the option to drop the player or put them on the bench. Goals (G) and assists (A) are the number of goals and assists they have scored during the season. Plus/minus (+/-) is a stat that is unique to hockey. Every time they are on the ice during an even strength situation (both teams have the same number of skaters), their +/- can be changed. If their team scores a goal during this situation, they get one added to their +/-, while if the other team scores during htis situation then one gets subtracted from their +/-. The higher the +/-, the better the player (usually... players on good teams have high +/- too). Shots (S) is the number of shots that they have taken (shots are good), and PIM (penalties in minutes) is the number of minutes they have spent in the penalty box (PIM are bad). PPP is "power play points", which is the number of goals and assists they have scored when their team is on the power play (when the other team has a penalty). PPP are good, since they indicate that the player is good enought to be on his team's power play unit, and that they know how to score goals and assists. It also lists the

NHL team, position, and nightly opponent for each player.

5.5.3 Benched Players

The benched players list is exactly the same as the starting players list, except for a few things. First, the limit to the number of players is dependent on the number of players on the starting roster (each user can have ten non-goalies total). Second, the two "Action" options are Start and Drop, which add the player to the starting roster (assuming there is room) and drop the player from the team, respectively. It still lists the season stats for each player, but benched players do not contribute to their fantasy team's running totals for statistics

5.6 All Players vs. Available Players vs. Goalies

All Players lists every player in the mySQL database (which is every NHL player that has scored a goal or an assist), and their season stats for each of the six categories (see descriptions of each category in section 5.5.2). It also gives the option to Add, Drop, or Trade, depending on the status of the player. Available Players is exactly the same as All Players, except it only lists players that aren't on teams (so it only gives the option to Add them to the user's roster). The Goalies page lists every goalie that has played in at least one game, and gives their season stats for each goalie category (see section 5.5.1 for a description), and the option to Add, Drop, or Trade, depending on whether or not a user has claimed them. All three pages give the option to sort the players/goalies by each of the categories, and all three pages also list each player or goalie's NHL team, nightly opponent, and owner.

5.7 Logout

The last option available to a user is the Logout button which... logs them out.

6 Results

To put it simply, my project resulted in a working fantasy hockey league. It was fully functional, but a few feature that I had intended to code weren't added (I had a list of possible features, knowing that I wouldn't get to them all). These included trade and a message board. But the heart of the program (the data mining, the statistics calculation for each team, and the win-loss-tie calculator) is fully functional. Now that the regular hockey season is over, the final results are in. The winner was Ricky Biggs, with Ryan Comes and I just two points behind (two points is equal to one win or two ties).

In terms of knowledge, this project has resulted in me learning many things. In addition to learning PHP (I knew none before I started this project), I have learned to manage large databases and to deal with mySQL. I learned about data mining, and what is involved in extracting large amounts of data from a website. I also learned how to create a professional looking website that allows users to manipulate data. Overall, this project has resulted in a very worthwhile learning experience.

7 Conclusion

In conclusion, I had a very valuable year in techlab. I learned two new languages, PHP and mySQL, and I also sharpened my webpage building skills. I learned how to maintain an extemely large database with many users, and to respond to recommendations from my users. I would definitly recommend this project to future students, though they should probably start with what I have already done (they can finish it up, and then work on an artificial intelligence user, which would be a very good techlab project). I learned a lot about data mining, and how it is used in the real world, as well as how to incorporate it into my project. I am sure that the knowledge that I gained this year will serve me well at Olin and in the distant future.

8 Recommendations

8.1 Improvements

If someone were to do my project again, or if they were to do a similar project, I have some useful recommendations for them. The biggest problem that I had was time. The hockey season ended during third quarter, so there was nothing I could add to my program after that time. I then focused on my paper, and projects outside of Techlab for all of fourth quarter. I would recommend that if someone were to created a fantasy hockey league in a future class, they should be excused from the written report for the first semester. This would allow them to focus on their program while the season was still underway, and while

their new features could still be useful. Then in second semester, when the hockey season is winding down and there is less coding to do, they could really get to work on their report. This would be the best use of their time, since there would never be a class where they had nothing to do. They would have a decent program by the middle of the year, giving them the rest of the year to write an excellent report.

8.2 Future Projects

One problem with my project was that once I was done with the initial coding (during the first semester), I didn't have to code anything dealing with data mining. I wrote statparser.php early on in the year, and that was the only page that actually accessed ESPN.com and mined their hockey statistics. After that, I wrote pages for my fantasy hockey website that used the data that was being mined every night (I spent time with the layout of the page, and adding new featuers). I don't really see any way I could've incorporated more data mining into my project, but I've thought of another solution for future projects. Future projects dealing with data mining should include other computer science aspects as well. My original project was going to be to write a fantasy hockey program, and then to write an artificial intelligence user that would play it. This would have been a great project, but I ran out of time (my recommendation in section 8.1 would help alleviate the time crunch). If the report was postponed until second semester, then it would give students time to develop their project during the firs semeseter, and then they could incorporate more computer science aspects. The artificial intelligence player would have been a great addition to my project, but during

fourth quarter (when I actually had time to write it), the hockey season was already over.

9 References

Chawla, Sanjay. University of Sydney. "Introduction to Data Mining." 2004.

http://www.it.usyd.edu.au/~chawla/5318/lecture1.pdf

Data Mine, The. 2000. http://www.the-data-mine.com/

Mansour, Steve. Sitescooper. "A Tao of Regular Expressions." 1999.

http://sitescooper.org/tao_regexps.html

Ramsay, Stephen. University of Virginia. "Using Regular Expressions." 2004.

http://etext.lib.virginia.edu/helpsheets/regex.html

Rea, Alan. The Queen's University of Belfast. "Data Mining." 2004.

http://www.pcc.qub.ac.uk/tec/courses/datamining/stu_notes/dm_book_2.html

Stein, Bob. VisiBone. "Regular Expressions." 2004.

http://www.visibone.com/regular-expressions/