Predicting the Outcomes of Squash Matches

Overview

Data on a number of junior squash matches and players was gathered and used to predict the outcomes of other junior squash matches (not in terms of win/loss, but in terms of point differential). The end goal was to find which factors are most predictive of point differential, and then devise a predictive metric for it. To do this, PCA was used to determine which factors contributed most to good predictions, and then a regression was run using the principal components.

The reason to use point differential and not the binary win/loss metric was that it allowed for more variation and detail in the outcomes—a match with a point differential of 2 is very different than a point differential of 15, but by win/loss standards, both would be coded as 1. Point differential is also highly related to match outcome—a favorable point differential of 5 or more leads to a 99.79% chance of winning the match, and a point differential of just 1 still leads to a 95.85% chance of winning the match.

Metrics Collected

For each player in the match at question:
Previous match outcomes (W/L)
Previous match point differentials
Home city
For each match:
Ranking difference between players
Rating difference between players
Difference in date of account creation
Match round (i.e. round 1, semifinals, etc.)

Data Collection

This was far and away the most difficult and time-consuming aspect of this project. The data was gathered from the US Squash website and then cleaned and processed into a useable format. To do this, a simple web scraper on the website to gather the necessary data with a number of steps would have been ideal. However, after much experimentation and frustration with various tools and libraries for doing this, it was determined that it was nearly impossible because of the way the website was set up (mostly because the data was contained in a restricted iframe on the page, not the page itself, but also for other website architecture reasons). In order to view the source for that iframe, one must replace ussquash.com in each page url with modules.ussquash.com, which loaded the iframe and showed the source code. However, a web scraper still would not work on this, so the individual source code was pulled (for some metrics there was a relevant line in the code that to grab, but other times all lines were necessary) for each relevant page. Additionally, nearly all of the information gathered was password-protected, making all of these processes longer and more complicated.

Another difficult thing about getting this data was that it was not displayed neatly on each page for each match. In order to limit the scope of the matches the model was trained and tested

on, just one certain type of tournament (called JCT, or Junior Championship Tour), in addition to the national championship, for the 2016-17 season, was used. This yielded a total of five tournaments with about 700 played matches in each. In order to get a list of all of the matches played, the HTML of the full results list from each tournament page was downloaded and then cleaned into a useable format. Below is a brief explanation of all of the various metrics collected and how they were collected. Note: this isn't really related to the outcome, but it is included for the sake of thoroughness. Reading it will provide context on the metrics of the project, but isn't required for understanding the results.

Previous match outcomes: Because each player's match history contains a unique set of circumstances under which the matches are played (based on which types of tournaments they play, teams they are on, etc), it did not make sense to get each player's full tournament history. Additionally, match history is located not on one page but on successive pages with a set number of results each, which would drastically increase computation time. So, the match history statistic was limited to simply other matches in the same tournament, or matches in the other four tournaments considered in the dataset. The single nice thing about the website is that it is organized by player ID number, and player names rarely appear without an ID number attached in the HTML. Player home pages are accessed by ID number rather than name, making them easier to iterate through. Then the records of each player in each tournament were compared to get a full list of each player's matches. I had to be careful because the matches after cleaning were listed [winner, loser, loser scores[1]]. Thus, each match had to be searched through twice for each player: once by the winner and once by the loser.

Previous match point differentials: Once the data described above was generated, the last column of the match array was used to calculate point differential. In essence, each value was appended to the score of the other player (11 or greater), and then the list was split by player, and then the point differential calculated. Basically, [7, (8), 11, 3] would go to [[11, 7], [8, 11], [13, 11], [11, 3]] which would go to [11, 8, 13, 11], [7, 11, 11, 3], and then to 43-32 equals a point differential of 11.

Ranking: US Squash provides all junior players with a certain number of played tournaments a ranking based on their outcomes in those tournaments; different finishing positions in different levels of tournaments provide various amounts of "points," which are averaged. This data was obtained from each player's home page. For most people it is password protected, which made it more difficult to get; however, having a password provided access. The real challenge was that the player's ranking changes over the course of the year as these tournaments in question occur. Each player's historical rankings were found, and then the algorithm used the corresponding ranking from the time of each match. Finding and gathering this data was long

---

[1] Winner scores are always assumed to be 11 in each game (games are played to 11), unless the loser's score is 10 or greater in which case the winner's score is two greater than the loser's score. So, a loser's score of [7, 8, 10] would indicate a match that went 11-7, 11-8, 12-10.
However, it isn't that simple—since matches are the best of five games, many matches consist of more than three games. The score would then be indicated with the games won by the loser displayed by the loser of that game (the overall winner)'s score. For example, a match that goes 11-7, 8-11, 11-2, 11-3 would be [7, (8), 2, 3].

and computationally expensive, and in the end it may have been faster to use a brute-force approach.
Rating: In addition, US Squash also calculates a rating based on some undisclosed factors. Unfortunately, like ranking, it changes over time. Fortunately, historical ratings are listed pretty much side-by-side with historical rankings, so these could be obtained together.
Match round: This is pretty simple—the played matches come by round.
Age of account: While the accounts are not timestamped upon creation, player ID numbers are given out in order, so while you can not tell how long the two relevant accounts have been around for, you can take the difference of the two ID numbers to see which one was created first and approximately how much earlier.

Predictive Algorithm - PCA

PCA was used to increase the variance between the points of the data. Why not just run a regression without doing this? Because running a regression on this data wouldn't (and doesn't) produce helpful results at all.

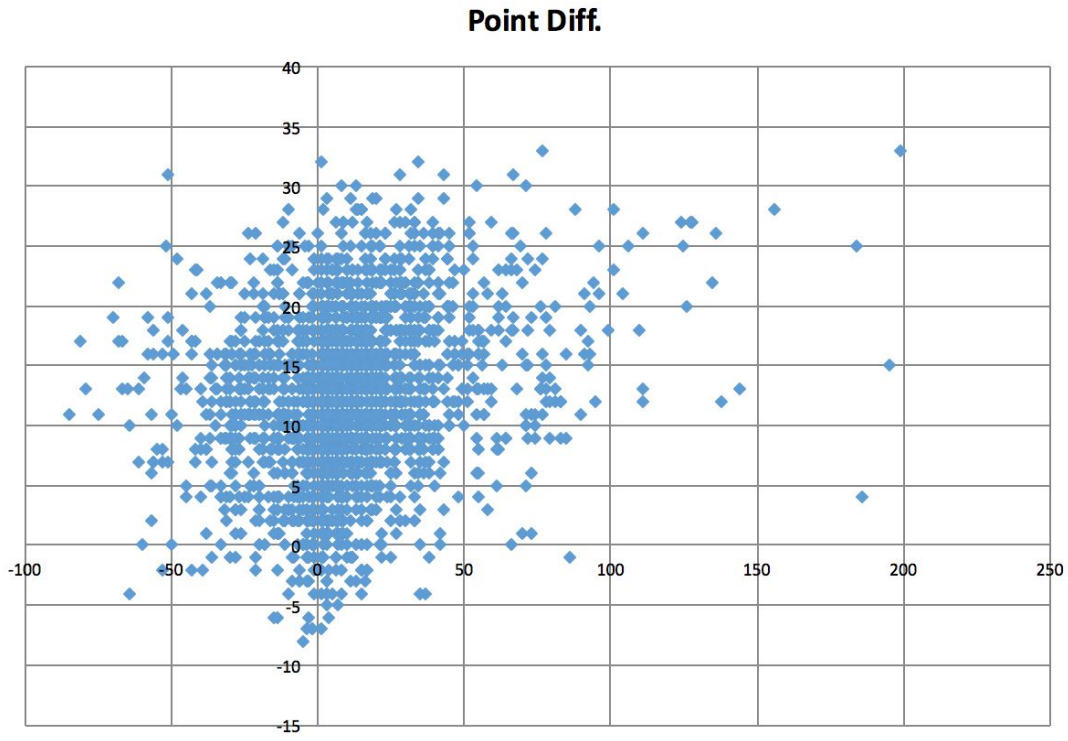

Chart 1: Ranking differential vs. Point differential

Below is the list of correlation coefficients for the various metrics. The end goal is to predict the point differential, so metrics that align closely with point differential will be most helpful in determining the result. Point total is most strongly correlated (-0.803) with point differential, which makes sense (although is not obvious)—the higher the point total (i.e. the more points are played), the closer the match is, and thus the lower the point differential. Matches that aren't very close have lower point totals overall and a higher point differential. However, it was

then obvious that the ratings and rankings included were not very useful in their current form, so they were adjusted to be difference in ranking and difference in rating, instead of winner/loser rankings and winner/loser ratings (since the rankings and ratings of the players were not useful unless the context of the other player's rating or ranking was known).

```
Correlation Coefficients, using the observations 1 - 2377
(missing values were skipped)

  TimeDifference        PointTot      Outcomes_w      Outcomes_l
          1.0000          0.1384          0.0101          0.1610 TimeDifference
                          1.0000         -0.1542          0.1828 PointTot
                                          1.0000          0.4181 Outcomes_w
                                                          1.0000 Outcomes_l

         AvgPD_w         AvgPD_l       Ranking_w       Ranking_l
          0.0054          0.1674          0.0504         -0.1792 TimeDifference
         -0.1858          0.2220          0.0678         -0.1748 PointTot
          0.9196          0.3928         -0.4005         -0.2119 Outcomes_w
          0.4051          0.8462         -0.2185         -0.3518 Outcomes_l
          1.0000          0.3896         -0.4291         -0.2237 AvgPD_w
                          1.0000         -0.2338         -0.4013 AvgPD_l
                                          1.0000          0.1803 Ranking_w
                                                          1.0000 Ranking_l

        Rating_w        Rating_l       PointDiff
          0.0053          0.0332         -0.1737 TimeDifference
          0.0123          0.0500         -0.8031 PointTot
          0.1338          0.1054          0.1676 Outcomes_w
          0.0876          0.1180         -0.2063 Outcomes_l
          0.1473          0.1021          0.2022 AvgPD_w
          0.1216          0.1565         -0.2608 AvgPD_l
         -0.1429         -0.1023         -0.0496 Ranking_w
         -0.0662         -0.1249          0.2272 Ranking_l
          1.0000          0.1309         -0.0343 Rating_w
                          1.0000         -0.0858 Rating_l
                                          1.0000 PointDiff
```

Chart 2: Correlation Coefficients 1

Making that change yielded the correlation results below. Now, ranking difference is more significant, but still not nearly as important as point total.

```
Correlation Coefficients, using the observations 1 - 2377
(missing values were skipped)

5% critical value (two-tailed) = 0.0402 for n = 2373

  TimeDifference        PointTot      Outcomes_w      Outcomes_l
          1.0000          0.1384          0.0101          0.1610 TimeDifference
                          1.0000         -0.1542          0.1828 PointTot
                                          1.0000          0.4181 Outcomes_w
                                                          1.0000 Outcomes_l

         AvgPD_w         AvgPD_l     RankingDiff      RatingDiff
          0.0054          0.1674         -0.1929         -0.0073 TimeDifference
         -0.1858          0.2220         -0.1995         -0.0223 PointTot
          0.9196          0.3928          0.0614          0.0211 Outcomes_w
          0.4051          0.8462         -0.1787         -0.0255 Outcomes_l
          1.0000          0.3896          0.0689          0.0177 AvgPD_w
                          1.0000         -0.2135         -0.0338 AvgPD_l
                                          1.0000          0.0494 RankingDiff
                                                          1.0000 RatingDiff

       PointDiff
         -0.1737 TimeDifference
         -0.8031 PointTot
          0.1676 Outcomes_w
         -0.2063 Outcomes_l
          0.2022 AvgPD_w
         -0.2608 AvgPD_l
          0.2352 RankingDiff
          0.0380 RatingDiff
          1.0000 PointDiff
```

Chart 3: Correlation Coefficients 2

To go over each metric and how it relates to point differential (because the above chart is difficult to read, given the signs):
Time difference: The more recently the loser's account was created relative to the winner's, the higher the point differential (0.1737)
Point total: The higher the point total, the lower the point differential (0.8031)
Winner's/Loser's outcomes: The more wins the winner has, the higher the point differential, and vice versa (0.1676, 0.2063)
Winner's/Loser's avg. point differential: The higher the winner's point differential, the higher the point differential in the match, and vice versa (0.2022, 0.2608)
Ranking difference: The greater the difference in ranking in favor of the winner, the higher the point differential in favor of the winner (0.2352)
Rating difference: A greater difference in rating in favor of the winner makes a marginal difference in ending point differential (0.038)

The main conclusion from this step was that rating is a surprisingly terrible predictor of match outcome. Although it was expected to be about as good as ranking, but it seems to have very little predictive value.

Here is the summary of the principal components:

```
Principal Components Analysis
n = 2120 (dropped 257 incomplete observations)

Eigenanalysis of the Correlation Matrix

Component  Eigenvalue  Proportion   Cumulative
    1       3.2067       0.3207       0.3207
    2       1.6374       0.1637       0.4844
    3       1.0727       0.1073       0.5917
    4       0.9069       0.0907       0.6824
    5       0.8678       0.0868       0.7691
    6       0.7772       0.0777       0.8469
    7       0.6704       0.0670       0.9139
    8       0.6242       0.0624       0.9763
    9       0.1532       0.0153       0.9916
   10       0.0835       0.0084       1.0000

Eigenvectors (component loadings)

                    PC1      PC2      PC3      PC4      PC5      PC6      PC7
TimeDifference    0.094   -0.371    0.150    0.831    0.126   -0.012   -0.303
PointTot          0.034   -0.552   -0.113   -0.350    0.104   -0.230   -0.544
Outcomes_w        0.445    0.337    0.127    0.102   -0.011    0.080   -0.175
Outcomes_l        0.437   -0.263    0.145   -0.216    0.019    0.337    0.080
AvgPD_w           0.447    0.355    0.117    0.109   -0.001    0.053   -0.135
AvgPD_l           0.439   -0.300    0.076   -0.221    0.019    0.265    0.102
Ranking_w        -0.297   -0.262    0.134    0.139   -0.013    0.631    0.314
Ranking_l        -0.292    0.285   -0.013   -0.136    0.086    0.562   -0.650
Rating_w          0.132    0.058   -0.680    0.091    0.686    0.125    0.142
Rating_l          0.136   -0.073   -0.654    0.153   -0.703    0.157   -0.074

                    PC8      PC9     PC10
TimeDifference   -0.182    0.005   -0.004
PointTot          0.445   -0.025   -0.031
Outcomes_w        0.375   -0.006    0.696
Outcomes_l       -0.266   -0.693   -0.027
AvgPD_w           0.338    0.046   -0.716
AvgPD_l          -0.262    0.716    0.026
Ranking_w         0.555    0.027   -0.021
Ranking_l        -0.257    0.048   -0.018
Rating_w          0.036   -0.029    0.011
Rating_l          0.006   -0.024   -0.006
```

Chart 4: Summary of Principal Components

The top principal components were fairly representative of the data, particularly the first one with an eigenvalue of 3.207 (proportion of 32.07%). Since this principal component described a significant amount of variation in the data, a two-axis regression with point differential and results of PC1 was run. Here's a small sample of what the full table looks like, with the individual values under each principal component:

PointDiff	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
-1	0.1322271	-2.4810939	0.61406289	0.57106722	0.10170964	1.76418638	0.00328631	-0.3734417	-0.4735983
14	2.28411584	-0.2231547	-0.462435	0.02351887	-0.8625651	0.00080913	-0.3585522	0.16799986	0.36367243
21	2.54339859	1.43054608	-0.9492916	-0.4391102	-0.5141551	1.7388426	0.23080307	-0.8757511	-0.423922
12	3.3943857	-0.3790879	-0.2636098	-0.1766261	0.41841538	0.84963237	-0.0575231	-1.1410187	0.2367054
17	2.62998882	-0.5087638	-1.74449	-1.2342253	0.33171997	-0.225034	-0.2303498	0.94198322	0.01275055
20	5.62397681	0.16053222	-0.5464569	-0.033435	-1.0040145	0.28140931	0.23326703	-1.3105906	0.95734429
23	1.92867652	1.61819888	-0.8372696	-0.2485883	-1.0327599	1.02457766	0.26543317	-0.2035143	-0.3573665
9	-2.6522982	-1.2321032	0.01252717	0.07484668	-0.0382445	0.33517823	0.7695835	0.3686359	-0.4276719
20	0.12794677	1.57868291	-0.3931117	0.15886384	-0.8629801	0.18018014	0.10725194	-0.3234505	-0.3291462
8	-1.5582171	-0.0366007	-0.2398615	0.00941317	-0.3397699	0.65517683	-0.6451518	0.1545835	-0.4280217
15	-1.0257931	0.17095273	-0.7469585	-0.3626648	0.93343833	-0.0495977	0.51262158	-0.3707031	-0.3468216
16	-2.210392	0.35839463	-1.4261572	-1.0009668	0.32162993	-1.1426203	-0.104222	0.10161462	-0.2871664
8	0.34270028	-0.7169794	0.03914969	0.48138783	-0.9089283	0.01821023	-0.5018556	-0.2640269	0.36577042
11	0.27616568	-0.702125	-0.9043605	-0.6199714	-0.0263887	-0.6784606	-0.3910208	-0.3835496	0.0861223

Chart 5: Sample of individual points under PCs

Plotting this data for the first principal component vs. the point differential, like the regression data, still looks useless and clumpy (a tiny bit less so, but still):

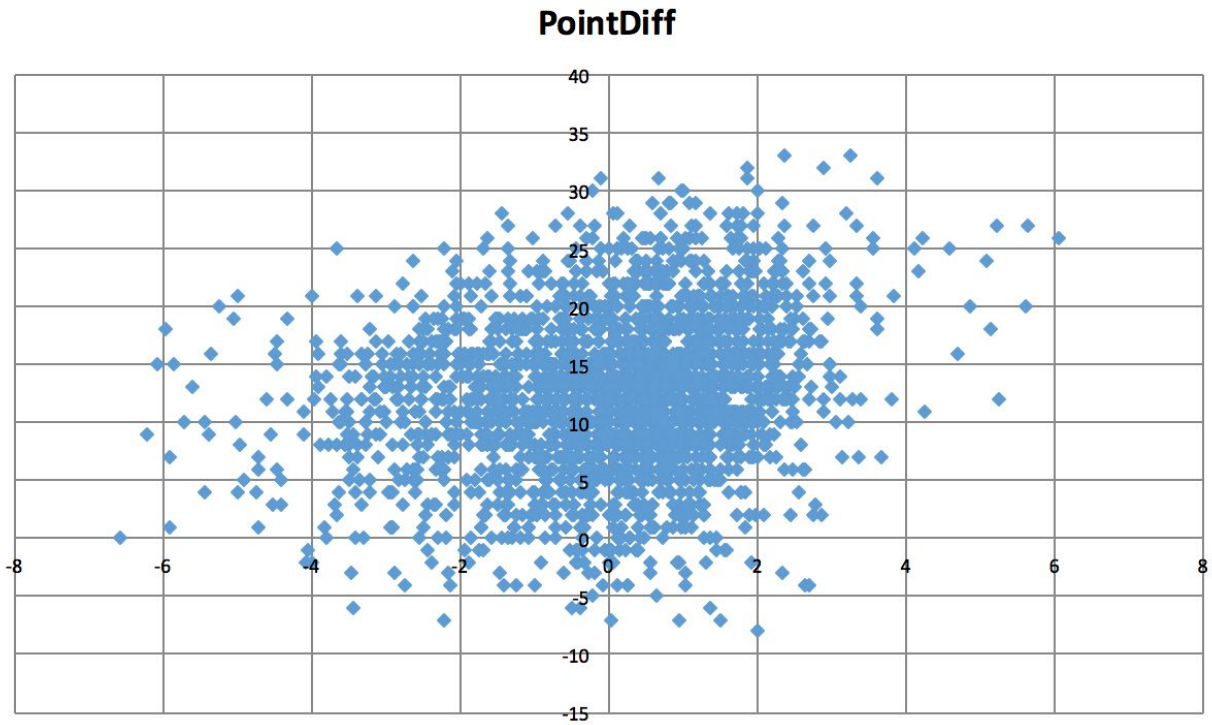

Chart 6: Graph of individual matches, PC1 vs Point Differential

However, running a regression along every principal component might be more effective. The regression returned these coefficients. A graph is not included, because it was a nine-dimensional regression.

PC1	0.889837
PC2	3.75586
PC3	−1.79443
PC4	−1.01708
PC5	−1.12301
PC6	−1.22615
PC7	4.76948
PC8	0.924020
PC9	0.226049

After obtaining this, the model was tested using a separate testing set.
This data doesn't help much unless we know how each individual point in our testing set fits into each of these principal components. Because we know from Chart 4 what each principal component eigenvector is, we can fit each new point into the PCA. Data from the PCs above and Chart 4 allowed us to do this with the help of a short program.

gretl was then used to look at the correlation between point differential and the new metric for the testing set, which returned a correlation of 0.9999, making in a ridiculously well-correlated metric. Even when a quite significant amount of noise was added (almost 25% of the original numbers) to the testing set numbers, the correlation remained at 0.9999. This was simply too good to be true—it was most likely because the size of the testing set was only about ⅛ the size of the training set. So, the entire process was repeated over again with a more reasonable comparison: a testing set the same size as the training set.

In the end, the metric was about 96% correlated, making it a great (if difficult to execute) predictor of match outcomes. In order to ease the computation of predicted point differential (so you don't have to go through the recreation of the PCA and then the regression), a short program that combined those two steps into one was used.

However, it was still possibly unclear if my metric really did outperform simpler methods. Its accuracy was compared to that of three other predictive measures:
a simple multivariate regression of all of the factors used in the PCA;
the rankings provided by US Squash;
the ratings provided by US Squash.
Here are the correlation results for all of these metrics.
New process: 0.96398
regression: 0.3539
ranking: 0.1611
rating: 0.0655

As you can see, rankings and ratings are indeed poor indicators of the outcome of the match, and barely beat out random guessing when it comes to predicting point differential. A 9-factor regression is much better, but still much worse than the created PCA metric.

Note: Here are some of the data files and data processing programs generated over the course of this project. There were too many of them to organize, so browse at your own risk. A next step would be to determine if “home-field advantage” is actually a factor—because many players travel long distances for these major tournaments, the effect on play would be interesting to analyze. That’s why there are a number of files related to locations in here. More of this can be seen here, under “What Even is ‘Squash’?”