Overview

The dataset I used for this assignment was all ATP tour match histories for Men’s singles tennis in 2012 form [1]. The data provides information far beyond the 4 majors (Grand Slams) consisting of 2594 matches played in 2012 in 63 different ATP tournaments including the majors. In order to use the match history data in NodeXL I first projected the Winners and Losers column to a set of unique players and used those as vertices. An edge is drawn between vertices for every match played between the two players.

![Figure 1 - Overview of Pro-tennis network](http://nodeXL.codeplex.com)
Figure 2 - Matches played at different rounds in all tournaments
Stiff Competition
The first step in generating Figure 1 was hand annotating a “Round” field in the dataset. While the data came with a “Round” attribute it was used as a free-text form where later rounds would be written as “3rd Round” or “Quarterfinals”. By using Excel’s Data -> Text To Columns tool I was able to parse out the round data for those with numeric options easily. I then had to go in and fix up all the late round matches by hand. After this data was available I could use NodeXL’s dynamic filters tab to show all the matches played at a specific round across all (or some subset of) tournaments.

Figure 1 how the network evolves into later and later round matches across all ATP tournaments played in 2012 (not just Grand Slams). The nodes are sized by the player’s average ranking throughout each match played in 2012. The image is quite dramatic highlighting how men’s singles is tournaments are dominated by a handful of players at the top ATP world rankings. In the very last image is hand cleaned version of the last round (7) in all major tournaments, colored by Series (e.g. Grand Slams which is shown in blue). The 4 Grand Slam finals played in 2012 have seen the same 4 players, Djokovic, Nadal, Federer, and Murray in the finals.

Upsets
Each edge in the dataset has the world ranking of both the winner and loser going into the match. By taking the ratio of Lrank and Wrank we get a measure for how dissimilar the two players are in rank. By using this computed ratio as the width of the edge we can begin to look for upsets by searching for large arrows in the graph. Using NodeXL’s dynamic filter I filter by edge width while also grouping players based on their average world rank in 2012. The arrows are colored by surface type, blue being hard court, purple being clay, and green being grass.

It interesting to note that no one upset the top 4 players Djokovic, Nadal, Murray, or Federer who are in the lower right box (which disappeared due to what I can only presume is a bug in NodeXL). Even the players inside the top 50 were rarely upset by anyone from the other groups. Most upsets came from the middle-tier players playing other middle-tier players (e.g rankings 50-500).
Importance of First set tie breaks

A tie break occurs in a tennis set when both players have won 6 games within that set. If it is the first set when the tie break occurs then both players are dead even, as far as score is concerned, at the beginning of the tiebreak. At the end of the tiebreak however, only 1 player is awarded the set giving an edge to the winner. By using the W1 (number of games won in first set by winner) and the L1 (number of games won in first set by loser) fields on the edges we can identify matches where there was a first set tiebreak (e.g. W1 = 7 AND L1 = 6 OR W1 = 6 AND L1=7). Figure 4 and Figure 5 highlight is just how important these tiebreaks are. The edge density is much higher for Figure 5 than Figure 4. You can also see that the largest loss in density comes from the bottom left group, which is the group of tennis players who maintained an average world rank throughout 2012 in the range [150-500]. This would indicate that for these players it is even more critical to win a tie-break than players in the other groups.
Figure 4 - Matches where loser wins first set tiebreak

Figure 5 - Matches where winner wins first set tiebreak
NodeXL Critique

While I found NodeXL to be a powerful tool in visualizing network data it did not come without its challenges. The first thing I noticed is that large networks were off limits. Some of my initial attempts at finding a network dataset were to import twitter data from various users. 1 such network (Rebecca Black’s) yielded 20,000 edges, a moderate network size, which degraded NodeXL’s performance so much that it was unusable despite a powerful desktop computer with 16 GBs of RAM.

Some other suggestions I noted down while working with NodeXL were:

- It would be nice to be able to set the arrow head size independently of the arrows body. This would have made it possible for me to see arrow directions while still zoomed out, without cluttering up the window with thick arrows
- Be able to select edges in the graph window as opposed to only vertices
- While I could create auxiliary numeric columns parsed from categorical columns it would be better if NodeXL allowed dynamic filters on categorical data directly that way I would not need to remember what the numeric <-> categorical mappings were

References

http://www.tennis-data.co.uk/

http://www.atpworldtour.com

http://nodexl.codeplex.com/