Consider a sports match between two entities, x\_1 and x\_2. Before the match, one can use all relevant information about the two entities to generate P(x\_1), a prior win-probability for x\_1. \_\_\_\_ cite how this has been carried out in other sports, prediction mechanisms, betting odds, etc. \_\_\_cite how this has been done in tennis, referencing papers you have read, eg the neural net paper from UK with 4% returns. Also reference the in-point prediction model with probability of points won on serve, updated throughout the match (how can we approach this from a Bayesian perspective?)

Now consider a time series g\_1,g\_2,…g\_n where g\_k contains all information about the match through time k. …formally explain the sub-modular property to my prediction model that must hold in all new methods I develop…

Throughout, I will employ random forests and neural nets, as well as other common tools from the sci-kit learn arsenal. While these models do not adhere to the above property, random forests make predictions based on randomly generated partitions of training data and neural nets adeptly express non-linear relationships between input variables and output, all models I explore in any depth will maintain this property. While it is the case that losing a point, game, or set can be beneficial to a player in some situations (see “tanking”, add a footnote), these situations are rare enough and against the spirit of tennis and the ATP’s rules of conduct. I think any model that tried to capture such instances would likely be overfitting, with few abstract characteristics to latch onto, especially given the sparseness of our datasets here.

Yet, nearly all papers concerning match prediction in tennis deal with pre-match forecasts. Building upon current models, I attempt to build a model that fits posterior in-match probabilities of the form P(x\_1|g\_k), where g\_k is the sequence of match events through time k.

Big thanks to the statistics department for letting me write a thesis on something I find truly interesting and have spent far too much time watching over the past eight years.

This research has mainly been fueled by personal interest and a desire to provide in-match win probabilities as an avid tennis viewer. In reality, an in-match forecasting system that outperforms the betting market does not even promise anyone a get-rich-quick scheme, as commissions and fees on sports betting websites are substantial. Still, I do hope that my work on this project can contribute to the body of knowledge surrounding tennis match predictions and the betting market. At the very least, I hope knowledge gleaned from this work can ultimately make markets a little more informed efficient, an endeavor I picked up from the fine men, young and old, of Susquehanna International Group LLP.

Links re James-Stein estimator:

--https://stats.stackexchange.com/questions/5727/james-stein-estimator-how-did-efron-and-morris-calculate-sigma2-in-shrinkag

--http://chris-said.io/2017/05/03/empirical-bayes-for-multiple-sample-sizes/

--https://stats.stackexchange.com/questions/119786/james-stein-estimator-with-unequal-variances

--perhaps also look into modeling in-match predictions with HMM models? Could easily tie this in as another machine-learning topic (see dissertation)

ultimately, we want models that will be hip to service break advantages and break points. We also want to differentiate between being up 2-0 and 5-3 in the final set. Currently, our model cannot differentiate between the two when it uses game differentials.

Ex. It is 1-1; 0-40 in the third set. Player 0 has three break points on the player 1’s serve. Given all the info we have about this match, we want to calculate Player 0’s probability of breaking serve, from this score. Then,

P(match win) = P(match win | Player 0 breaks) \* P(Player 0 breaks) + P(match win | Player 1 holds) \* (1-P(Player 0 breaks)).

This kind of model seems much more promising than the current logistic regression we have, although I’m sure logistic regression can be somewhat improved.

At end of intro: display summary statistics of elo, surface elo, win % from a set up, split, etc. with a bunch of graphics and examples (use the entire atp match database, all matches since 1990)

Weakness of logistic regression: display with an imwp graph of the Nadal-Muller epic at Wimbledon, given that it continually goes up every time Muller wins a game, doesn’t recognize break points as effectively

(later on: would be fun to produce IMWP graphs for Federer-Nadal 2008 Wimbledon and Djokovic-Federer 2011 US Open)

Would be promising to construct a player metric which encompasses their historical performance when leading a set, trailing a set, in a final set, etc. It’s a question of how many samples to require before deciding you have a valid estimate of their fortitude

Features to include in the future: court surface, surface-specific ELO ratings, clutch index, temperature

(--read some statistical papers on bootstrap sampling, other forms of random sampling)