Bayesian Hierarchical Models in Tennis

Peter Tea

10/04/2020

What are Point-Based Models?

Point-Based models can provide exciting and competitive insight into the game of Tennis. With them, we can answer questions like: Which team/player is favoured to win in a match-up? How unlikely is an observed upset?

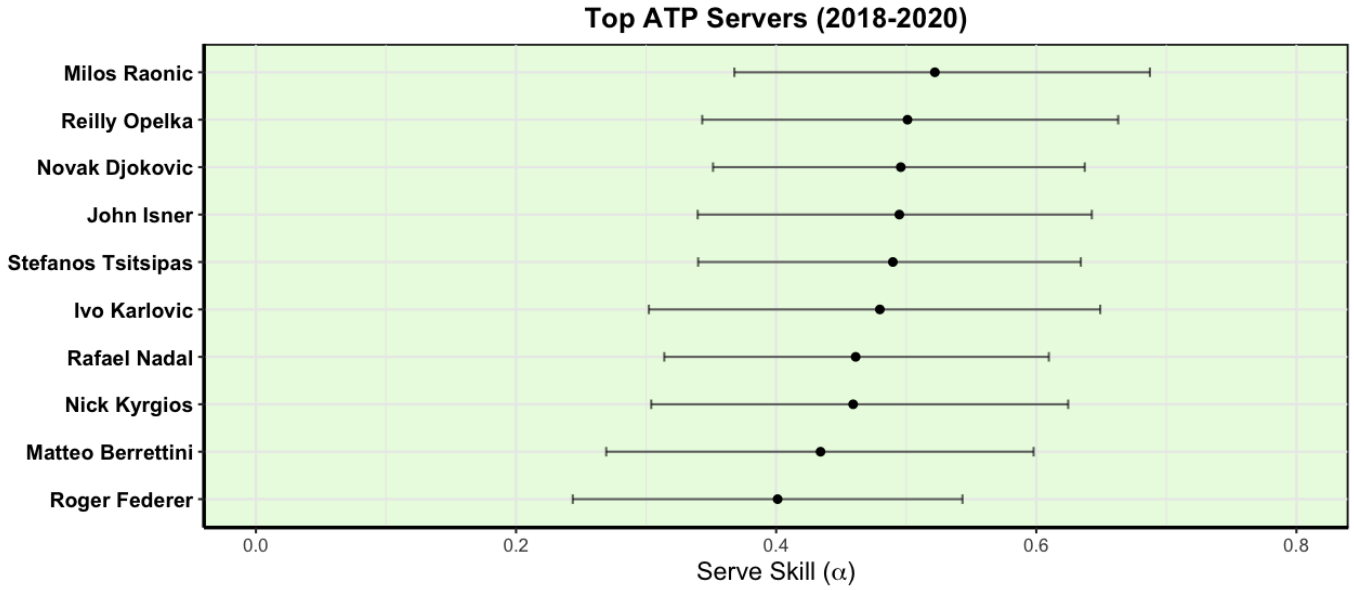
In constructing point-based models, we first make some assumptions on service point outcomes - the main one being that service points are I.I.D. for an entire match. This amounts to saying that all service points in a match are played equally. With the I.I.D. assumption, (Newton and Keller 2005}) showed (using combinatorics and geometric series representations) that we can calculate the probability of either player winning the tennis match and the probability of observing specific set scores by simply knowing both players' serve win probabilities.

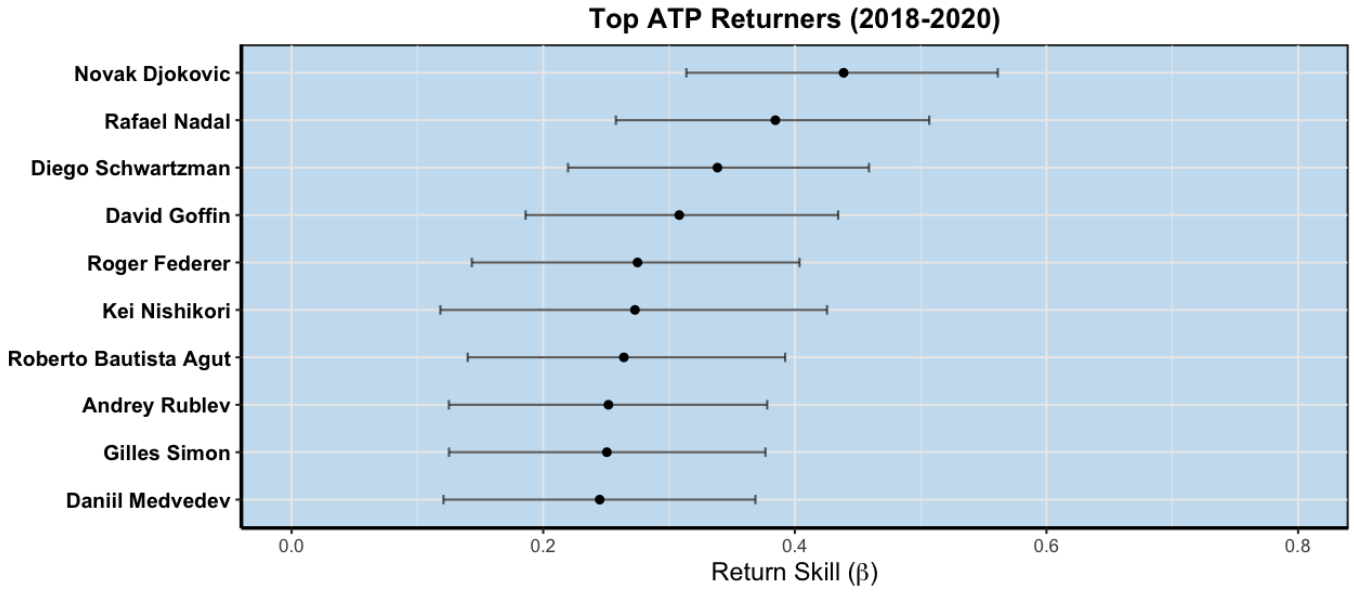
Recently, Ingram (2019) proposed a Bayesian Hierarchical Model to provide posterior serve win probabilities for players in a head-to-head (singles) matchup. This Bayesian model accounts for player serve and return skills fluctuating across different periods, as well as player-specific court preferences and the tournament at which the match is being played.

For this post, I present some interesting visualizations produced from fitting this Bayesian point-based model. Tennis data curated from 2018 to 2020 (until the end of the Australian Open) from one of Jeff Sackmann's repositories, was used to feed the model. While all necessary code to fit the model is already available on the author's GitHub page, I've instead fit the model in R (whereas the author uses Python). Bayesian posterior samples were drawn using `rstan`, a package that uses a Hamiltonian Monte Carlo algorithm.

Who are the top skilled ATP Players?

From the posterior distribution of player serve and return skills, I've ranked the top 10 players for each skill based on the players' posterior median. Below are plots representing the top 10 ATP players for serve and return skills (dots represent posterior medians, while bands represent a 90% credible interval).



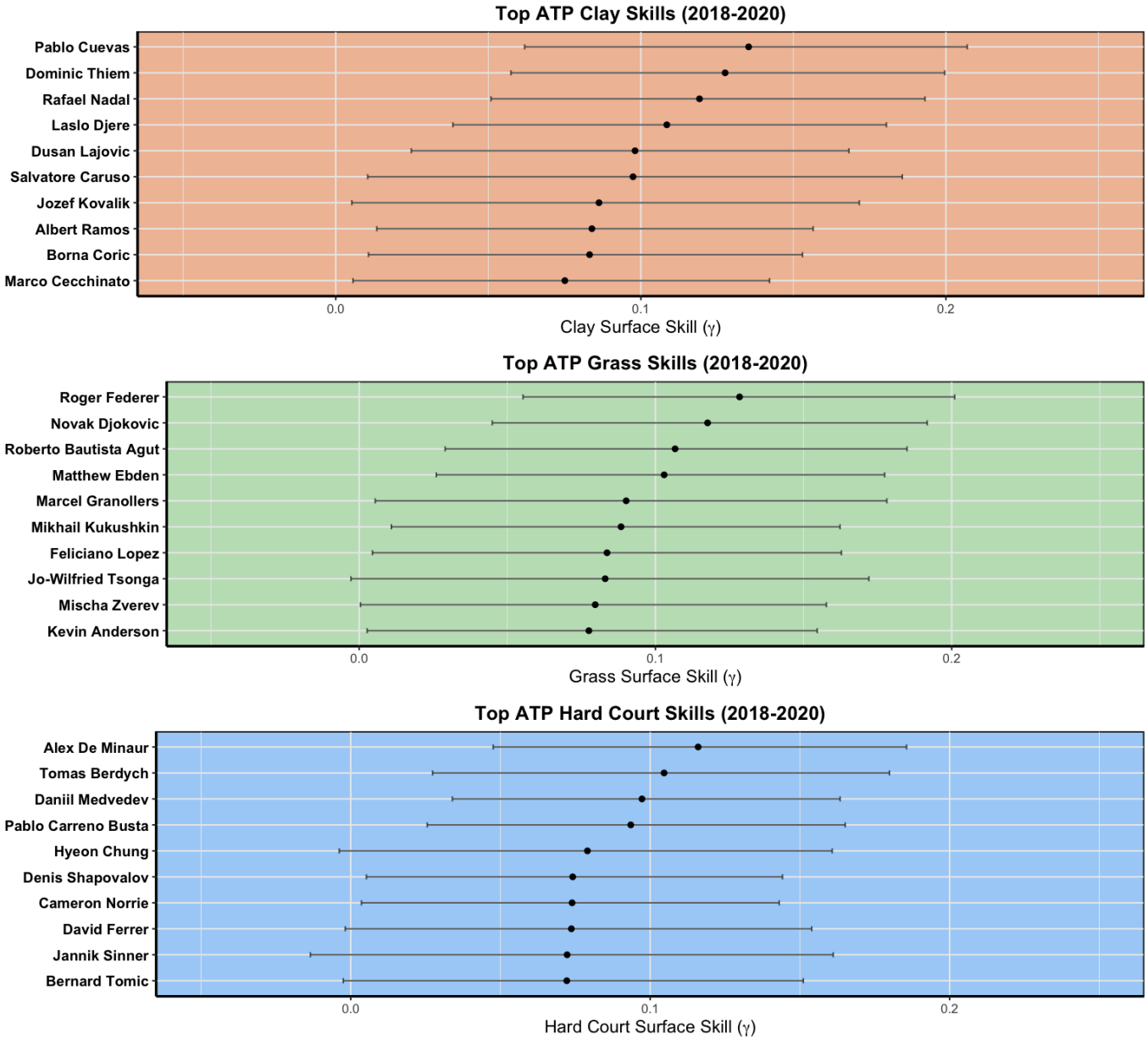


In general, players will have higher serve skills than return skills since it is easier for players to win points on their serve compared to their return against the opponent's serve. One surreal feature shown here is that Novak Djokovic's posterior median return skill is larger than some of the top 10 players' serve skill!

Unsurprisingly, Canadian Milos Raonic is crowned the ATP's top server while Novak Djokovic gets the crown as the ATP's top returner. Reassuringly, we do see that these rankings contain players who expect to be at the top (i.e. Nadal, Fed, Djokovic, etc.).

Who are the top surface ATP Players?

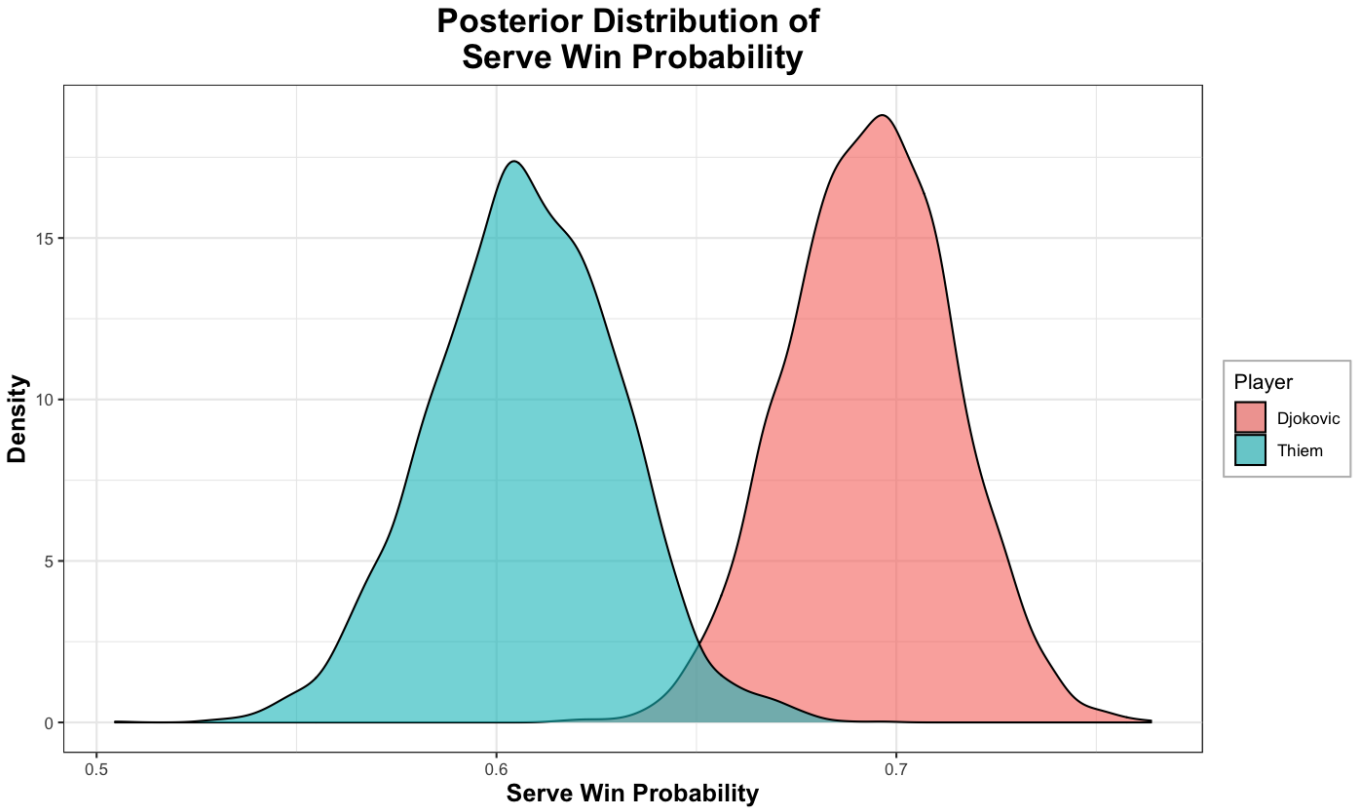
Next up, we'll look at which players get the biggest boost from the 3 tennis surfaces. The following plots should be interpreted as "additional surface skill", which were previously unaccounted for in the players' serve and return skill plots above.



From the clay plot, Rafael Nadal (i.e. the "King of Clay") easily cracks the top 10 players boosted from the clay surface list, as expected with his stellar career on the clay courts. From the grass plot, we round out the other 2 kings (Fed and Djokovic) who rank particularly high on this surface. Finally, for the hard-court plot, we find other top familiar names in the ATP. Surprisingly, David Ferrer (retired 2019) was ranked 8th on our list! Here, it wouldn't make too much sense to say that a retired player would still be ranked top 10. Perhaps this anomaly may be attributed to some deft performances on hardcourt from Ferrer before his retirement.

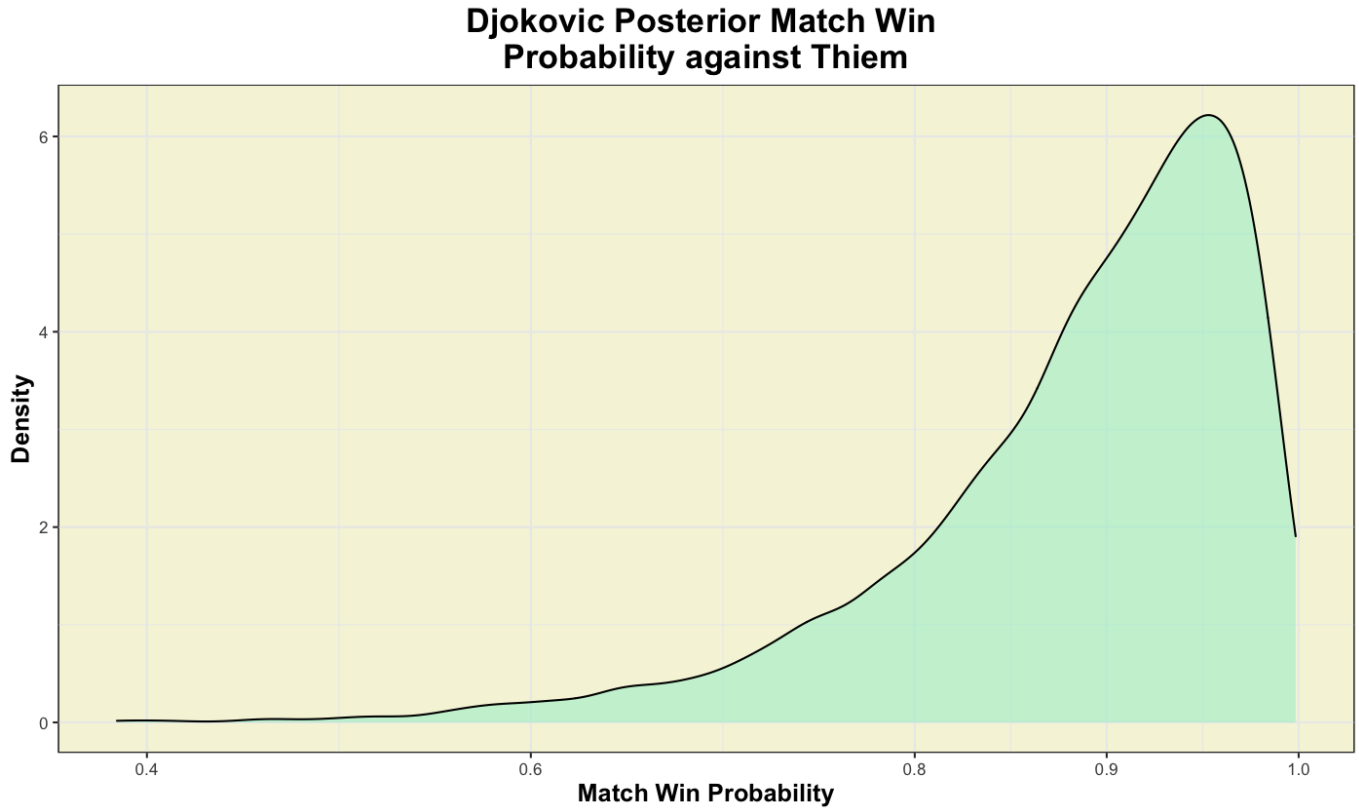
Djokovic vs. Thiem 2020 Australian Open

Next up, I used the model to predict the Australian Open 2020 Final match played between Djokovic and Thiem. For both these players, I first obtained their posterior serve win estimates.



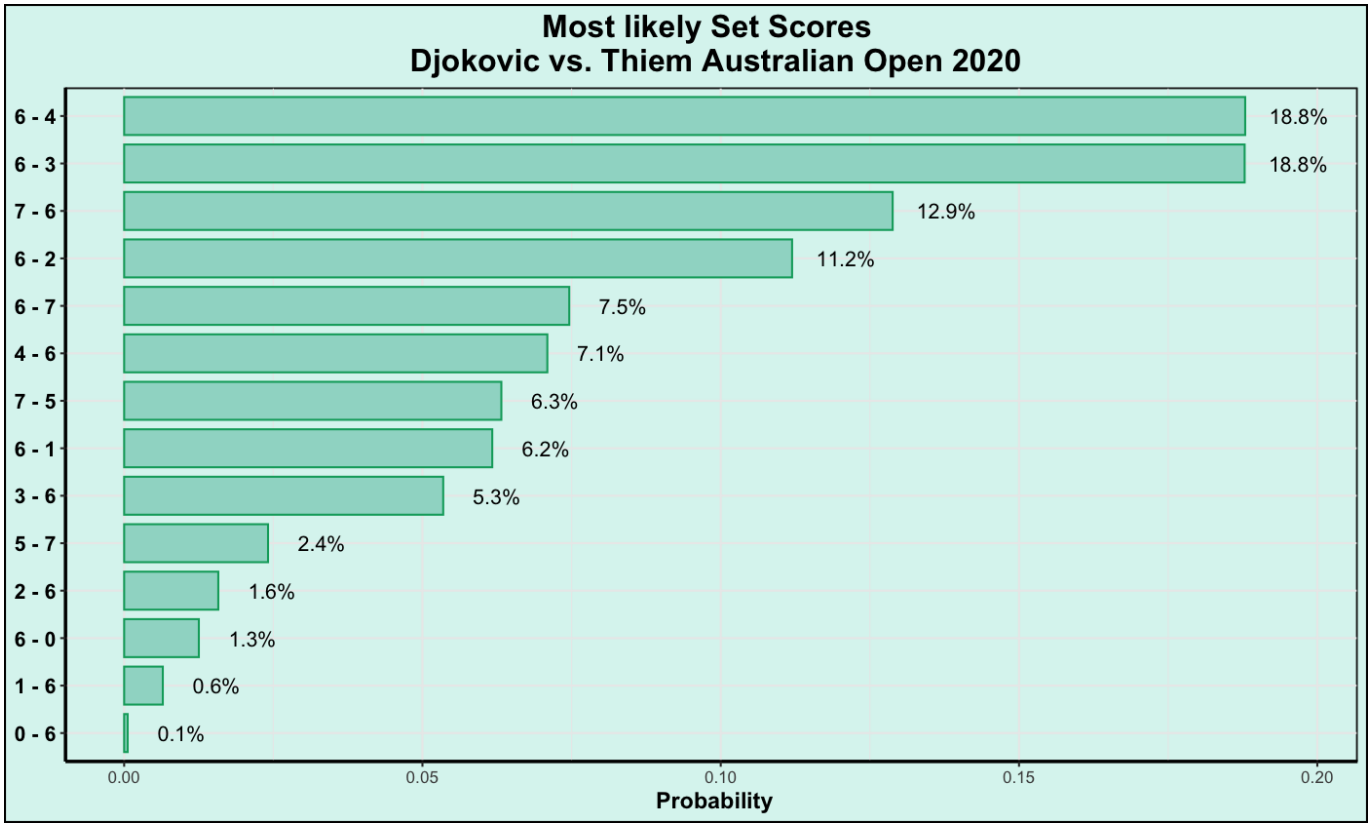
From the above Posterior serve win probabilities plot, we see that Djokovic (in red) has an advantage in winning his serve games against Thiem (in blue). When considering posterior medians, we would predict that Djokovic would win just under 70% of his service games, compared to Thiem winning just under 61% of his own respective service games.

Using the posterior serve wins for both these players in the head-to-head matchup, we can then produce the posterior distribution of either playing winning the match.



From the plot above, we calculate Pr(Djokovic Winning Match vs. Thiem) has 90% credible interval of [70.4%, 98.4%] - so quite a considerable advantage for Djokovic!

With the posterior serve wins for both these players in the head-to-head matchup, we can also produce the probabilities of observing set scores in this matchup.



The most likely set score, as predicted from the model, is 6-4 and 6-3 for Djokovic.

How do these predictions stack up with what already happened? Reassuringly, Djokovic did end up beating Thiem in this match with a final score of (6-4, 4-6, 2-6, 6-3, 6-4). These observed scores mostly adhere to the most likely scores we predicted (except for the 2-6 set score, which had a 1.6% chance of occurring!)

Who's got the best chance of winning the remaining Slams?

Unfortunately, Wimbledon has been cancelled and there will be no crowned grass champion this year. However, with the point-based models we can instead predict who has the best chances of winning at this illustrious grand slam tournament. By calculating each players' predicted serve win probability at Wimbledon, we get that `Novak Djokovic` has the highest serve win probability of any ATP player, and thus is predicted to be the most likely Wimbledon champion. Besides Djokovic, who has the best chance of winning Wimbledon? Using the model, I obtain match win probabilities for ATP players against Djokovic. Here are the top 10 players with the highest match win probabilities against Djokovic:

Player	Win %	90% Credible Interval
Rafael Nadal	23.7	[4 - 60]
Roger Federer	20.9	[4 - 56]
Milos Raonic	7.7	[1 - 29]
Roberto Bautista Agut	5.5	[0 - 30]
Stefanos Tsitsipas	3.7	[0 - 21]
Juan Martin del Potro	2.6	[0 - 23]
Matteo Berrettini	2.6	[0 - 16]
Jo-Wilfried Tsonga	2.0	[0 - 15]
Stan Wawrinka	1.3	[0 - 11]
Andrey Rublev	0.8	[0 - 11]

Based on the model, Nadal has the highest probability of beating Djokovic at Wimbledon (23.7 % with 90% credible interval of [4%, 60%]), followed closely by Roger Federer (20.9 % with 90% credible interval of [4%, 56%]).

Let's say (God forbid), that the remaining 2 grand slams are cancelled this year. Who would have the best chances of winning the remaining grand slams this season?

Starting with Roland Garros, unsurprisingly Rafael Nadal has the highest probability of winning service points at this tournament and is thus crowned another French Open title. How does the rest of the ATP stack up against Nadal at Roland Garros? I present the top ATP players with the highest win probabilities against Nadal at 2020 Roland Garros:

Player	Win %	90% Credible Interval
Novak Djokovic	28.1	[6 - 66]
Stefanos Tsitsipas	15.5	[3 - 46]
Dominic Thiem	9.8	[1 - 37]
Milos Raonic	6.7	[0 - 34]

Player	Win %	90% Credible Interval
Roger Federer	6.4	[1 - 31]
Matteo Berrettini	4.4	[0 - 23]
Andrey Rublev	2.9	[0 - 20]
Juan Martin del Potro	2.7	[0 - 21]
Alexander Zverev	2.4	[0 - 15]
Jan Lennard Struff	1.8	[0 - 13]

Here, we see that Djokovic has the highest probability of beating Djokovic at Wimbledon (28.1 % with 90% credible interval of [6%, 66%]), followed by Stefanos Tsitsipas (15.5 % with 90% credible interval of [3%, 46%]). Roger Federer falls to 5th in terms of players most likely to beat Nadal at Rolland Garros (behind Thiem and Raonic).

Ending out our hypothetical scenarios with the (usual) last grand slam tournament of the season, US Open, we find that Novak Djokovic has the highest probability of winning service points at this tournament and is thus crowned the US Open chip. How does the rest of the ATP stack up against Djokovic at US Open?

Player	Win %	90% Credible Interval
Rafael Nadal	28.3	[8 - 61]
Roger Federer	17.1	[3 - 46]
Stefanos Tsitsipas	17.1	[4 - 44]
Daniil Medvedev	10.6	[2 - 35]
Milos Raonic	9.4	[2 - 30]
Dominic Thiem	8.6	[1 - 29]
Juan Martin del Potro	8.4	[1 - 38]
Andrey Rublev	8.0	[1 - 29]
Alexander Zverev	5.7	[1 - 23]
Roberto Bautista Agut	5.6	[1 - 23]

Here, we see 4 ATP players with double digit probabilities of winning against Djokovic at US Open which suggests that this grand slam may be the most uncertain tournament to predict. Nadal and Federer appear to be the most likely culprits to defeat Djokovic with probabilities 28.3% and 17.1% respectively. It is very interesting to note that Milos Raonic was consistently ranked in the top 5 of players with the highest chances of winning a grand slam this season. At age 29, hopefully we will see some great things from the Canadian Ace as the season resumes!

Being up-front about the assumptions

All predictive models we make are wrong. We hope to adequately capture all underlying processes, however to fit models we have to make some simplifying assumptions. The main assumption made in fitting this model was that service point outcomes are I.I.D. for an entire match (i.e. all service points in a match are played equally). Of course, we know that the point outcomes in practice are not independent, nor are they identically distributed. Throughout a match, there exists momentum from game-to-game played which means that the outcome of a previous serve may influence the

current serve. Furthermore, the performance of players on serve may differ depending on the pressure situation of the match. We may have reason to believe that our performance on the first service point compared to when we face a break opportunity may be drastically different. In fact Klaassen and Magnus (2001) showed that while the I.I.D. assumption in tennis does not hold, the effects are quite weak and thus the I.I.D. assumption may be a "good approximation" for match prediction.

Another assumption made in fitting this model was that player serve and return skills follow some Gaussian random walk that change smoothly from one period to the next with equal variance. This assumption was made to account for the fluctuation of player skills varying through time. Some players may go through stretches where they play unbelievably well or unbelievably poor. Players also evolve their skills as they get accustomed to the ATP circuit, so we assume that player skills are changing from one period to the next. The Gaussian random walk assumption says that player skills are constant within a period, and changes when we jump into the next period. In fitting the model, I set each period to be 4 months long, meaning that player skills will be constant within this 4 month window, before changing as we enter the next period. However, this assumption may or may not be a true representation of how player skills evolve through time. For example, older players approaching retirement may exhibit drastic decline in skill which of course is not a smooth transition.

Data Issues

Some of the entries in the data taken from Jeff Sackmann's repository appear to be incomplete. Some matches indicate that 0 service points were won on 0 serve point opportunities, while other entries indicate numbers that don't make sense under a tennis match context. In a best of 3 set match, we would expect that a player should have at the very least (4 points / game) * (6 games / set) * (2 sets / Match) = 48 serves.

Some matches may have ended prematurely (ex: from player injury), and we see some entries with service games less than 48.

Conclusion

Using a Bayesian Hierarchical Model, we can account for individual player effects by considering their court surface preferences, as well as the evolution of their serve and return skills as the season progresses. Using Stan, we obtain posterior estimates of serve, return and surface skill for each player and saw who was likely to hoist the grand slam tournaments in 2020.

Appendix

Bayesian Hierarchical Models

Here, we discuss some useful components of Bayesian Hierarchical Models (HM) to help us better understand how this model operates. We also look at useful extensions of Bayesian HM and how they can be adapted in a sport context.

In practice, observational units may be sampled from clusters within a population. Under this scheme, responses from the same cluster could be more similar to each other than responses originating from two different clusters. In this case, Hierarchical models can be applied to account for the dependence structure in the collected data (Gelman 2007). In sports, we oftentimes have repeated measurements on the same team or player. Hence, Hierarchical models can be adapted in sports to account for the inherent differences between teams or players.

In Bayesian inference, we have that the posterior distribution is proportional to the likelihood function and the prior distribution as follows:

$$\underbrace{\pi(\theta|\text{ data }\,)}_{\text{posterior}} \propto \underbrace{f(\text{ data }|\theta)}_{\text{likelihood}} \times \underbrace{p(\theta)}_{\times\text{prior}}$$

To fit Bayesian HM models, samples from the joint posterior distribution of all unknown parameters are first obtained. Then, posterior estimates for each unknown marginal parameter are acquired by integrating through the joint posterior.

In the case of Bayesian hierarchical models we model hierarchical structures with extra hyperparameters. Hence, the posterior is a joint distribution of all unknown parameters. Below, we introduce a hyperparameter (Γ) to model a level of hierarchy as follows:

$$\underbrace{\pi(\theta,\Gamma|\text{ data }\,)}_{\text{posterior}} \propto \underbrace{f(\text{ data }|\theta,\Gamma)}_{\text{likelihood}}\underbrace{p(\theta|\Gamma)}_{\text{prior}} \times \underbrace{\gamma(\Gamma)}_{\text{hyperprior}} \tag{1}$$

When a prior distribution itself depends on other parameters, these parameters are called *hyperparameters*. Hyperparameters are considered to be random and have their own distributions, known as hyperpriors. These hyperpriors could also depend on other random variables, thus adding another layer of priors. This recurring process is known as a hierarchical models and is a sequence of conditional distributions where the parameters come from their own distributions. We draw from posterior distributions using Stan, a software package employing an efficient MCMC algorithm.

Notes on Hamiltonian Monte Carlo, NUTS and Stan

Hamiltonian Monte Carlo (HMC) is an efficient MCMC algorithm implemented in the software package, Stan. HMC introduces an auxiliary momentum variable, ρ in its proposal step which will move our particle in the chain. Using ideas from Hamiltonian dynamics, we push the particle away from the initial point, and neighborhoods that we have already explored. This allows the chain to more easily explore the target state-space (especially in high-dimensional spaces).

We define the Hamiltonian (i.e. the total energy of the system) as:

$$H(p,\theta) = U(\theta) + K(p); \quad \text{where}\, U(\theta) = -logp(\theta); K(p) = -logp(\rho|\theta)$$

Here, $U(\theta)$ is analagous to the **potential energy** while $K(p)$ is analogous to the **kinetic energy** in the system. We then generate proposals as follows

1. $\rho \sim \mathrm{N}(0, \Sigma_{dxd})$ where d is the dimension of θ. Note that generating ρ is independent of current parameter values).
2. Evolve the joint system (θ, ρ) via Hamilton equations

$$\frac{d\theta}{dt} = +\frac{\partial K}{\partial \rho}$$
$$\frac{d\rho}{dt} = -\frac{\partial U}{\partial \theta}$$

To solve the above Hamilton equations, we use a leapfrog integrator (a numerical integration algorithm) to solve the two-state differential equations. This algorithm usually requires 2 user defined parameters: number of steps L and step size ϵ.

$$\begin{aligned}\rho &\leftarrow \rho - \tfrac{\epsilon}{2}\tfrac{\partial V}{\partial \theta}\\ \theta &\leftarrow \theta + \epsilon \textstyle\sum \rho\\ \rho &\leftarrow \rho - \tfrac{\epsilon}{2}\tfrac{\partial V}{\partial \theta}\end{aligned}$$

Here, we make half-step updates of the momentum, ρ, and full-step updates of the position θ.

Normally, the HMC is sensitive to the choice of the 2 user defined parameters, L and ϵ. Thankfully, the NUTS (No-U-Turn Sampler) algorithm chooses the values of these 2 parameters, obviating the need for user input. NUTS will find its step size (during the burn-in period) and also adaptively sets number of steps parameters on its own. The "No U-turn" comes from how the NUTS algorithm chooses L, which is made by ensuring that the current position continuously moves away from the starting position of the chain.

After each iteration we obtain a new proposal, obtain (p^*, θ^*) with corresponding Metropolis acceptance probability of $\min\left(1, \exp\left(H(\rho,\theta) - H\left(\rho^*, \theta^*\right)\right)\right)$.

In the end, HMC draws from a joint density,

$$p(\rho,\theta) = p(\rho|\theta)p(\theta)$$

however we can easily integrate through to obtain the marginal density of θ.

To illustrate the efficiency of NUTS HMC against other MCMC algorithms, here's a figure of a simulation study taken from (Hoffman and Gelman (2011)).

From Figure 7, we see that the target distribution is high dimensional (250 dimensions). The NUTS HMC appears capable of exploring the majority of the target state space efficiently, while the Random Walk (R-W) Metropolis Hastings gets stuck in local areas and is unable to explore the extremities of the target state space. More information on Stan and its employed algorithms can be found in (Carpenter et al. 2017) and (Betancourt 2017).

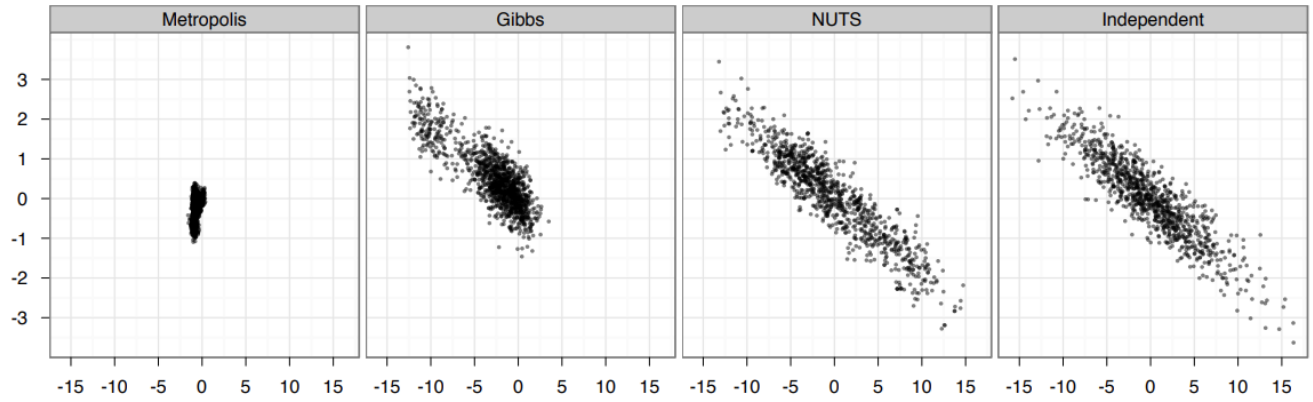


Figure 7: *Samples generated by random-walk Metropolis, Gibbs sampling, and NUTS. The plots compare 1,000 independent draws from a highly correlated 250-dimensional distribution (right) with 1,000,000 samples (thinned to 1,000 samples for display) generated by random-walk Metropolis (left), 1,000,000 samples (thinned to 1,000 samples for display) generated by Gibbs sampling (second from left), and 1,000 samples generated by NUTS (second from right). Only the first two dimensions are shown here.*

Figure 1: Image from (Hoffman and Gelman (2011))

Likelihood

For a given tennis match i, let n represent the number of service games played by a server, and y represent the number of service games won. Under the proposed model, we assume that the number of service games won (y_i) by a player in match i is given by:

$$y_i \sim Binomial(n_i, \theta_i) \tag{2}$$

where θ_i represents the the player's serve win probability. Under this construct, we assume that each outcome of a player's serve throughout the match is independent and identically distributed. The serve win probability is further modelled by the following logistic equation:

$$\text{logit}\left(\theta_i\right) = \left(\alpha_{s(i)p(i)} - \beta_{r(i)p(i)}\right) + \left(\gamma_{s(i)m(i)} - \gamma_{r(i)m(i)}\right) + \delta_{t(i)} + \theta_0 \tag{3}$$

where

- $\alpha_{s(i)p(i)}$: server s(i)'s serving skill in period p(i)
- $\beta_{r(i)p(i)}$: returner r(i)'s returning skill in period p(i)
- $\gamma_{s(i)m(i)}$: server's additional skill on surface m(i)
- $\gamma_{r(i)m(i)}$: returner's additional skill on surface m(i)

- $\delta_{t(i)}$: adjustment to the intercept at tournament t(i)

- θ_0 intercept representing the average player's probability of winning service point

Intuitively, the model implies that a player's serve win probability is dependent on the player's serve skill adjusted by the opponent's return skill, the difference in the player's and their opponent's court surface skill and the tournament venue. Surface skills are important to capture, since we know that players have preferences for different court surfaces. In the extreme case, Rafael Nadal (i.e. the "King of Clay") has an absurd 98.5% win percentage on clay, and 12 French Open titles to boot. While his resume on hardcourt and grass are impressive, they pale in comparison to his dominance on clay.

Gaussian Random Walk, Priors and Hyperpriors!

Player serve and return skills are assumed to follow a Gaussian random walk to account for the fluctuation of these skills over time. This component helps model player serve and return skills evolving from one time period to the next. Initial serve and return skills are first drawn as follows:

$$\alpha_{.1} \sim N\left(0, \sigma^2_{\alpha 0}\right) \quad \sigma_{\alpha 0} \sim H(0,1) \tag{4}$$

$$\beta_{.1} \sim N\left(0, \sigma^2_{\beta 0}\right) \quad \sigma_{\beta 0} \sim H(0,1) \tag{5}$$

From above, initial skills follow some common distribution for all players, whose variance is governed by an informative hyper-prior. Skills with high variance are very unlikely, hence the choice of a half-normal hyperprior distribution. In the next time period, p + 1, the player's evolving serve (α_{p+1}) and return (β_{p+1}) skills are modelled based on their previous skills in period p as follows:

$$\alpha_{p+1} \sim N\left(\alpha_{.p}, \sigma^2_{\alpha}\right) \quad \sigma_{\alpha} \sim H(0,1) \tag{6}$$

$$\beta_{p+1} \sim N\left(\beta_{.p}, \sigma^2_{\beta}\right) \quad \sigma_{\beta} \sim H(0,1) \tag{7}$$

Lastly, tournament intercepts and surface preferences are drawn as follows:

$$\gamma_{..} \sim N\left(0, \sigma^2_{\gamma}\right) \quad \sigma_{\gamma} \sim H(0,1) \tag{8}$$

$$\delta_{.} \sim N\left(0, \sigma^2_{\delta}\right) \quad \sigma_{\delta} \sim H(0,1) \tag{9}$$

The setup of this model accounts for the lack of independence in serve performances observed for different players. Rather than fitting a separate model for each player individually, the proposed model accounts for hierarchy and considers skills from different players to have some shared characteristics. That is, we partially pool all players together and assume that all their skills come from similar distributions. Hyper-priors (hierarchical priors) for the variance of skills are introduced to account for the hierarchical levels in the model.

Bibliography

Betancourt, Michael. 2017. “A Conceptual Introduction to Hamiltonian Monte Carlo,” January.

Carpenter, Bob, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. “Stan : A Probabilistic Programming Language.” *Journal of Statistical Software* 76 (1). United States.

Gelman, Andrew. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models / Andrew Gelman, Jennifer Hill.* Analytical Methods for Social Research. Cambridge University Press.

Hoffman, Matthew D., and Andrew Gelman. 2011. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo.”

Ingram, Martin. 2019. “A Point-Based Bayesian Hierarchical Model to Predict the Outcome of Tennis Matches.” *Journal of Quantitative Analysis in Sports* 15 (4). De Gruyter: 313–25.

Klaassen, Franc J G M, and Jan R Magnus. 2001. “Are Points in Tennis Independent and Identically Distributed? Evidence from a Dynamic Binary Panel Data Model.” *Journal of the American Statistical Association* 96 (454). Taylor & Francis: 500–509. http://www.tandfonline.com/doi/abs/10.1198/016214501753168217.

Newton, Paul K., and Joseph B. Keller. 2005. “Probability of Winning at Tennis I. Theory and Data.” *Studies in Applied Mathematics* 114 (3): 241–69.