University of Cambridge

MPhil MLMI

4F13 Assignment 2

Candidate Number:
J902G

Date:
19/11/2021

Word Count: 1004

Part A

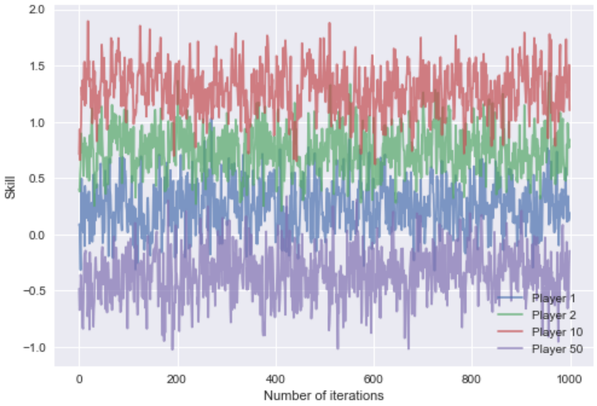

Figure 1: 4 samples of players' skills taken using Gibbs sampling over 1000 iterations.

Figure 1 shows 4 player skill samples from Gibbs sampling as a function of the iteration number. For the early iterations, it is often the case (and can be seen particularly for the red sample) that the initial samples are far from the equilibrium distribution mean, due to random sample initialisation. To account for this we use a 'burn-in' time, defined as the time taken for the samples to oscillate in their equilibrium pattern. For the samples in this practical this appeared to be small, and was set to 20 iterations.

Figure 2 (left) shows the autocorrelation of 5 samples as a function of the iteration lag-time, with the mean for all 107 samples on the right. Each sample appears correlated with the samples up to approximately 6 iterations before, due to the statistical dependencies of samples with previous iterations for the same player caused by using conditional distributions in Gibbs sampling. This can also be seen in Figure 1, where zoomed-in, the samples appear correlated, but zoomed-out, appear to fluctuate randomly around the equilibrium mean.

For these reasons, future Gibbs sampling discludes the first 20 iterations due to burn-in, and then records only every 10 (rounded up from 6 due to smaller autocorrelation standard deviation) iterations to account for autocorrelation. Even with burn-in and autocorrelation lags, 1000 iterations is sufficient to produce 98 robust samples for each player (see Figure 3 (left)).

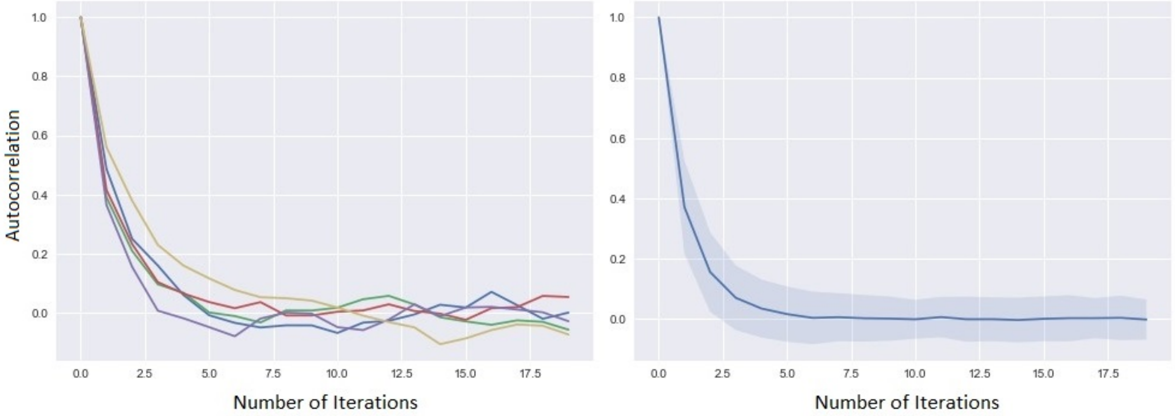

Figure 2: Plots of the autocorrelation between sampled skills as a function of the number of iterations. (Left) using 5 sampled players, (right) the mean over all 107 players with a 95% confidence interval.

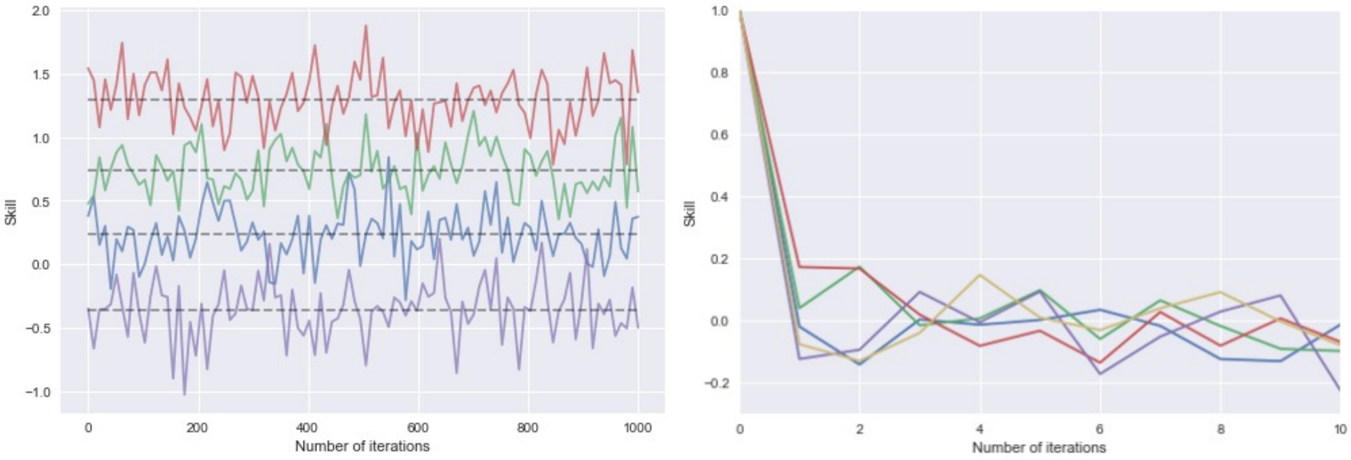

Figure 3: (Left) The same 4 samples as in Figure 1, but now with burn-in and autocorrelation times accounted for. (Right) The autocorrelations of the same 5 samples as in Figure 2, but now in this reduced scheme, with clear independence from the previous iteration's sample.

```
for p in range(M):
    m[p] = t.T.dot(np.array(p == G[:, 0], dtype=int) - np.array(p == G[:, 1], dtype=int))
```

Listing 1: Code snippet used to calculate the mean vector for each player's index in Gibbs sampling, using the difference of dot products between the performances, $\vec{t}$, and Boolean arrays (expressed as vectors of elements 0, 1 for subtracting) in order to provide a fast way to sum over indicator functions.

```
iS = np.zeros((M, M))
for g in range(N):
    I_g, J_g = G[g][0], G[g][1]
    iS[I_g][I_g] += 1
    iS[J_g][J_g] += 1
    iS[I_g][J_g] += -1
    iS[J_g][I_g] += -1
```

Listing 2: Code snippet for building the sum of precision matrices over all games in Gibbs sampling, exploiting the fact that only 4 entries are non-zero for each game, so we do not need to iterate over matrix indices.

Part B

In Gibbs sampling we are trying to converge to the intractable joint distribution over skills by taking multiple samples. In contrast, message passing and Expectation Propagation (EP) assumes that the distributions they are converging to are Gaussian, and converge to the approximate Gaussian posteriors of the marginal skills given the game outcomes. Convergence of the Gibbs sampling algorithm is discussed in *part a* with burn-in time correcting for random initialisation. The convergence of the means and variances of the approximate Gaussians in message passing can be seen in Figure 6.

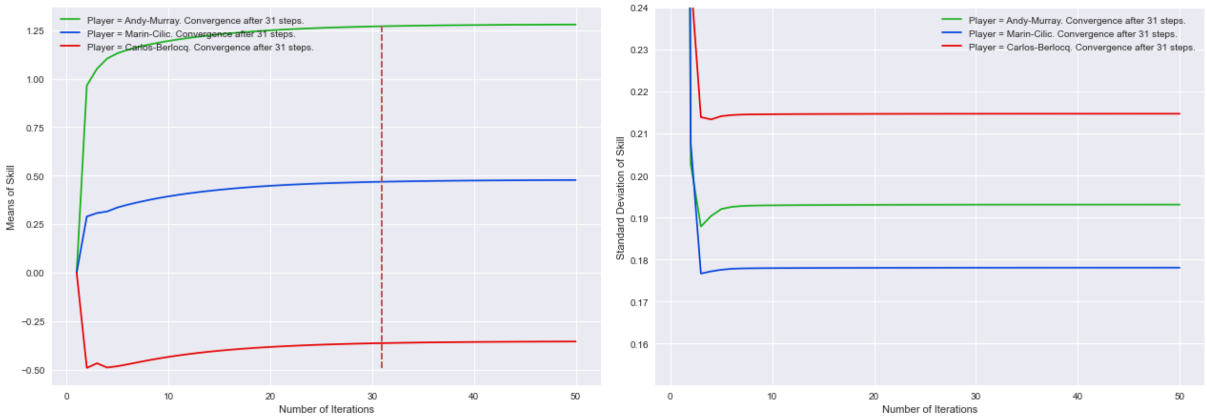

Figure 4: (Left) convergence of the means of the marginal player skills (found using message passing and EP for varying numbers of iterations) for 3 players, (right) convergence of the standard deviations for the same players. The method for judging convergence as described in Equation (1) assumes that there is not a subsequent change after initial convergence (i.e. that the convergence is always smooth), and thus the number of iterations before convergence was restricted to being greater than 10 (as before this there is non-smooth convergence as seen in the figure).

Convergence of the message passing algorithm is achieved when the approximate marginal Gaussian distributions do not change in future iterations, such that for each player j at iteration t:

$$q^{t+1}(w_j) - q^t(w_j) = 0 \tag{1}$$

If the above is true, then the means and variances of the approximate Gaussians would also not change. To allow for very small numerical differences, convergence was taken as when the difference between iterations is less than a tolerance (rather than exactly 0), which was set to 0.001. With this, convergences of 31 steps were found for the means of all 3 players in Figure 6 (at which points the standard deviations had also converged).

Part C

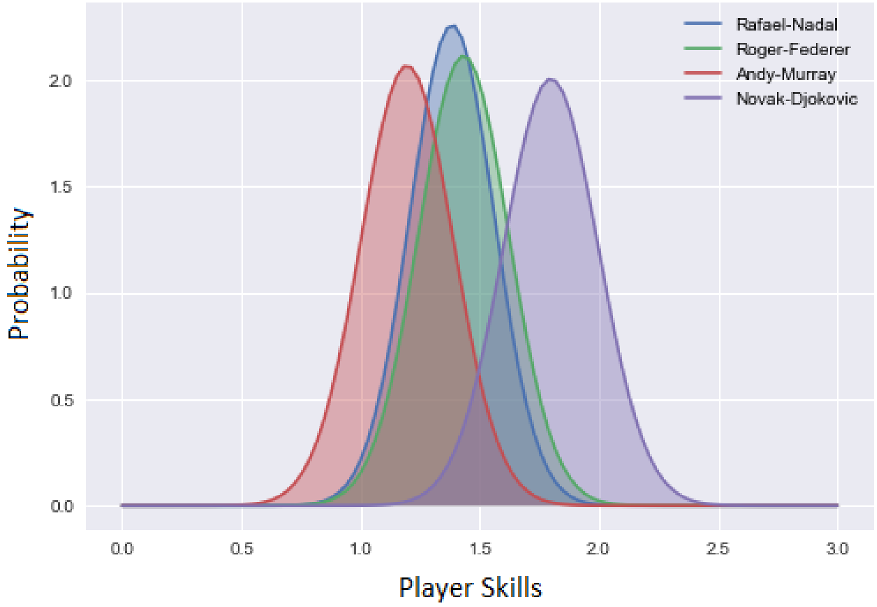

Figure 5: Gaussian density functions for the marginal skill distributions for the top 4 players, with means and variances found from message passing and EP. The plot qualitatively visualises why the probability of Federer's skills being higher than Nadal's is a lot closer to 50% than Murray's being higher than Djokovic's, for example; due to the closer means giving much greater overlap.

Table 1 shows the probability that the row players have higher skills than the column players calculated by approximating marginal skill distributions as Gaussians (as plotted in Figure 5), with means, μ_{player}, and variances, σ^2_{player}, approximated via message passing and EP. The probability of player 1's skill, w_1, being greater than player 2's, w_2, is calculated as:

$$\begin{aligned} P(w_1 > w_2) &= P(w_1 - w_2 > 0) \text{ (with } P(w_{player}) = \mathcal{N}(w_{player}; \mu_{player}, \sigma^2_{player})) \\ &= \int_0^\infty \mathcal{N}(w_1 - w_2; \mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2) d(w_1 - w_2) = \Phi\left(\frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}\right) \end{aligned} \tag{2}$$

(Higher Skills Probabilities)	Nadal	Federer	Murray	Djokovic
Nadal	–	42.6%	76.4%	6.0%
Federer	57.4%	–	81.0%	9.1%
Murray	23.6%	19.0%	–	1.5%
Djokovic	94.0%	90.9%	98.5%	–

Table 1: Probabilities that the players in the rows have higher skills than the players in the columns, produced by approximating the marginal skill distributions as Gaussian with mean and variance found by message passing and EP.

Conversely, table 2 shows probabilities of the row player *winning a match* against a column player, using the same marginal skill means and variances to above. This includes 'on-the-day' noise with variance 1 in addition to the variance of the skills difference distribution (see Equation (3)). Therefore, although the binary 'win'/'lose' is the same as the 'higher skill'/'lower skill' across the two tables, the former's variance is greater, and thus the probabilities are less polar and nearer to 50% than the skill difference probabilities, which is more reflective of actual game outcomes.

$$
\begin{aligned}
P(y=1) &= P(w_1 - w_2 + n > 0) \text{ (with } P(n) = \mathcal{N}(n;0,1)) \\
&= \int_0^\infty \mathcal{N}(z;\mu_1-\mu_2,\sigma_1^2+\sigma_2^2+1)dz \\
&= 1 - \Phi\left(0;\mu_1-\mu_2,\sqrt{1+\sigma_1^2+\sigma_2^2}\right) = \Phi\left(\frac{\mu_1-\mu_2}{\sqrt{1+\sigma_1^2+\sigma_2^2}};0,1\right)
\end{aligned} \tag{3}
$$

(Winning Probabilities)	Nadal	Federer	Murray	Djokovic
Nadal	–	48.1%	57.2%	34.4%
Federer	51.9%	–	59.0%	36.2%
Murray	42.7%	41.0%	–	28.1%
Djokovic	65.6%	63.8%	71.9%	–

Table 2: Probabilities that the players in the rows win the match against the players in the columns, produced by approximating the game outcomes as Gaussian, with equal mean to in 1, but with a higher variance due to a noise term representing uncertainty around *on-the-day* factors.

```
top4names = [’Novak-Djokovic’, ’Rafael-Nadal’, ’Roger-Federer’, ’Andy-Murray’]
top4 = [i for i, player in enumerate(W) if player in top4names]
higherSkillProb = np.zeros((4,4))

for i, P1 in enumerate(top4):
    for j, P2 in enumerate(top4):
        if P1 != P2:
            print(W[P1] + ’ has higher skills than ’ + W[P2])
            higherSkillProb[i][j] = 1 - norm.cdf(0,
                                     loc = mean_player_skills[P1] - mean_player_skills[P2],
                                     scale = np.sqrt(1/precision_player_skills[P1] +
                                         1/precision_player_skills[P2]))
```

Listing 3: Code snippet for calculating the probabilities that player 1 gets a higher score than player 2 when approximating the marginal skill distributions as Gaussian, and using the mean and variance from message passing and EP. To calculate the probability of player 1 beating player 2, the only mathematical difference is to add 1 to the variance within the np.sqrt() term (see Equation (3)).

Part D

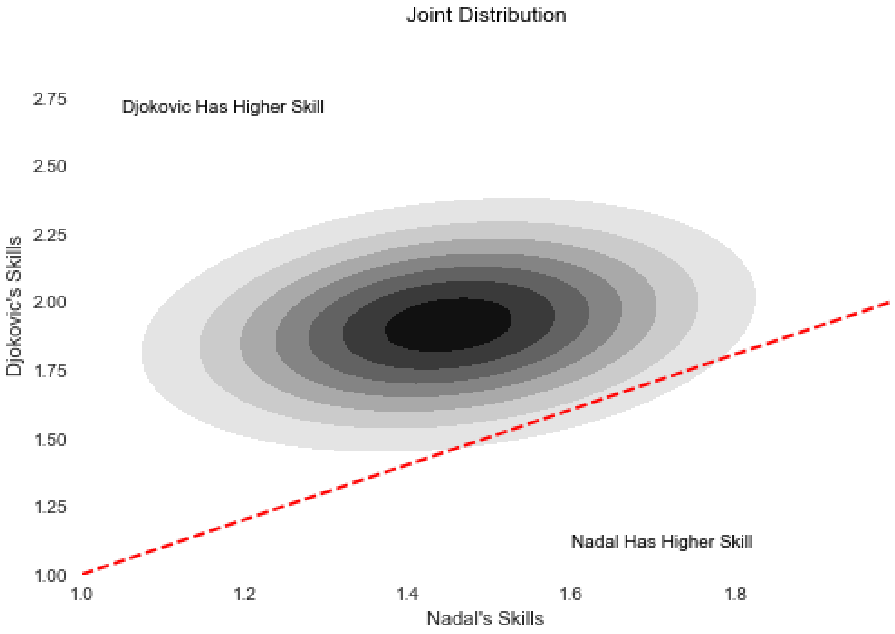

Figure 6: Plot of the approximate skill joint distribution for Nadal and Djokovic. The red line represents the skill difference boundary, and probabilities of higher skills are calculated as 2D integrals of the respective areas.

Table 3 was produced using Gibbs sampling and three different skill approximations. The first is that the marginal skills can be assumed to be Gaussianly distributed with independence between player skills, so that the probability that Nadal has higher skills than Djokovic is calculated using Equation (2) with Nadal as player 1 and Djokovic as player 2.

Then assuming joint Gaussian skills, a similar method is used, but now including covariance between player skills. The joint distribution is plotted in Figure 6, and the probability that $w_N > w_D$ is the area of the bottom right corner, given by:

$$P(w_N > w_D) = \int_{-\infty}^{\infty} \int_{-\infty}^{w_N} \mathcal{N}\left(\begin{bmatrix} w_N \\ w_D \end{bmatrix}; \begin{bmatrix} \mu_N \\ \mu_D \end{bmatrix}, \Sigma_{ND} \right) dw_D dw_N \tag{4}$$

The final approximation is simply counting the frequency of samples where Nadal's skill is higher than Djokovic's.

(Higher Skills Probability)	Nadal Higher Skill	Djokovic Higher Skill
Gaussian Marginal Skills	5.6%	94.4%
Gaussian Joint Skills	3.8%	96.2%
Directly Sampled Skills	3.2%	96.8%

Table 3: Probability that each player has higher skills, calculated using different approximations for the skills distributions.

From Table 3, the joint Gaussian skills probabilities seem similar to the direct counts of skill samples, whereas the marginal skills probabilities are less polar, with greater probability of Nadal having higher skills. The most accurate method is to use the joint distribution, because it does not assume that players' skills are independent of each other like the marginal skill Guassians (which is unrealistic as players improve from playing each other, so they have covariance), and it is more accurate than the direct sampling for few samples as Monte-Carlo approximations are less accurate for low sample numbers. However, the distribution of raw samples will eventually become more accurate as it approaches the *true* (intractable) distribution when the number of samples tends to infinity, but for small sample numbers, the joint Gaussian distribution is taken to be best.

Therefore, Table 4 shows the higher skills probabilities for the top 4 players using the approximate joint skill distribution and Gibbs sampling.

(Higher Skills Probabilities)	Nadal	Federer	Murray	Djokovic
Nadal	–	35.7%	73.2%	3.8%
Federer	64.3%	–	80.1%	5.4%
Murray	26.8%	19.9%	–	2.0%
Djokovic	96.2%	94.6%	98.0%	–

Table 4: A table representing the probabilities that the row player has higher skill than the column player, computed using Gibbs samples and by approximating the joint skills as Gaussians. It gives similar results to Table 1 (using message passing and EP), but with slightly more polar probabilities.

```
i_higherskills_j = np.zeros((4,4))
mean = [np.mean(nadal), np.mean(federer), np.mean(murray), np.mean(djokovic)]
std = [np.std(nadal), np.std(federer), np.std(murray), np.std(djokovic)]

def integrand(x1,x2,mean,cov):
    return mv.pdf([x1, x2], np.array(mean), cov)

for i in range(4):
    for j in range(4):
        if i != j:
            _mean = [mean[i], mean[j]]
            _cov = np.cov([nadal, federer, murray, djokovic][i], [nadal, federer, murray,
                djokovic][j])
            i_higherskills_j[i][j] = 1 - dblquad(integrand, -np.inf, np.inf, -np.inf, lambda x1:
                x1, args=(_mean, _cov))[0]
```

Listing 4: Code snippet used to calculate the 2D integral over the joint distribution for the probability of player 1 having higher skills than player 2, using Gibbs samples for the mean and covariance

Part E

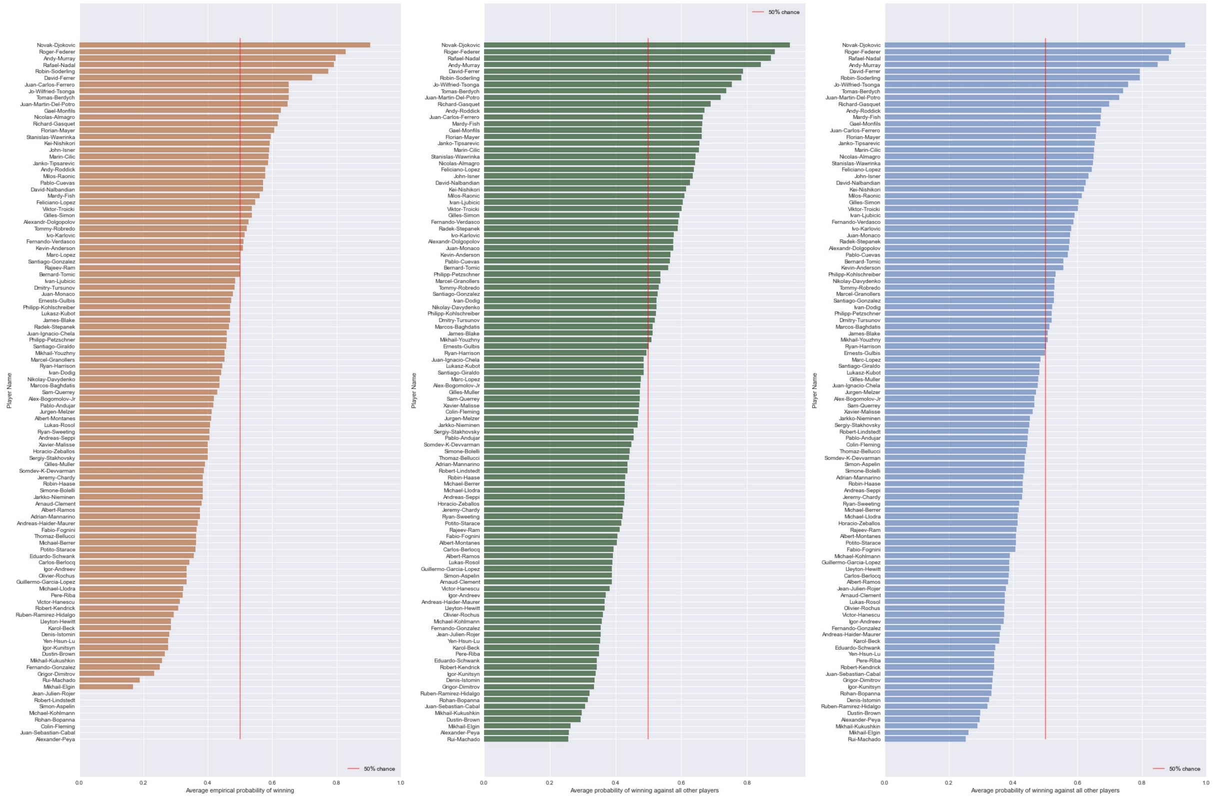

Figure 7: Rankings of the 107 tennis players based on a simple ratio of empirical wins to matches (left), average probabilities of winning over all other players from Gibbs sampling alone (middle), and average probabilities of winning over all other players with message passing and EP (right).

Figure 7 shows three variants for the rankings of the 107 players based firstly on empirical probabilities of winning, and then mean probabilities of beating all other players using Gibbs sampling and then message passing with EP. The rankings vary slightly, although the two probabilistic models only differ by one or two places in the rankings, and typically agree on top and bottom players, suggesting agreement between message passing and Gibbs sampling.

The greatest difference is between the probabilistic models and the empirical model. The short-comings of the empirical ranking are it's bias to the number of played games and that it does not account for the *skill* of the other players that each player plays. For example, a player being beaten by Djokovic has the same effect as being beaten by a much lower ranked player, and thus the rankings are skewed negatively for lower ranked players (see Table 5). This explains why many players who are ranked last in the empirical plot are ranked much higher in the probabilistic ones, which account for opponent skill.

	Peya	Cabal	Fleming	Bopanna	Kohlmann	Aspelin	Lindstedt	Rojer
Gibbs Ranking	105	101	56	100	88	81	65	90
EP Ranking	103	96	61	99	78	64	59	83

Table 5: Discrepancies between the empirical and probabilistic rankings. These players were all ranked joint last in empirical rankings due to no wins, but have very different rankings using the probabilistic models once their opponents skills are taken into account.