Mean container and precision matrix are computed in the following loop shown in Figure 1.

All iterations of skill samples for all players are collected to form a matrix of dimension 107x1100, which the ith column describes the skill samples of all players in ith iteration. The means and variances of all samples of all players are found in all iterations shown in Figure 2.

From 1st column of Figure 3, if each iteration of skill sample is from the same posterior distribution, the sample should be oscillating according to the mean and variance of the distribution. So, the gibbs sampler is able to move around the whole posterior distribution.

To find the auto-covariance coefficients of each player, the row vector (1x1100) representing the player’s skill samples over all iterations is put into the xcov function which outputs the coefficient vector (1x201) as the coefficients shown in Figure 4. And the sum of all coefficients in the vector gives the mixing time for each player.

Comparing all these sums from the players in Figure 5, the peak mixing time is 5.7. If the player has the sum of lower than or equal to 1, it means every sample has to be sampled to achieve independence. But even though a higher mixing time is used to sample this player, his samples are still independent. So, the peak mixing time should be taken. And to divide 1100 iterations completely, so every 5th Gibbs sample is kept to achieve independent sampling.

Burn-in describes the practice of throwing away some iterations at the beginning of an Markov Chain Monte Carlo.

When zooming in the samples in figure 5, there are always the first 4 samples far away from the equilibrium distribution. So, they are the burn-in samples.

For the gibbs sampler, the convergence occurs when the samples from conditional posterior can approximate the joint posterior distribution. It converges when the mean and variance of the samples are stabilized.

From 1st and 2nd columns of the figure 3, the mean of the player’s skill begins to stabilize quite early, around 50th iteration. But the variance of the skill continues to flunctuate until 300th iteration. So the required iterations should be 300.

In the message passing algorithm, it converges when the messages propagations can no longer update and change the distribution. From the figure 6 above, the skills of the players are getting updated continuously from the 1st to 20th iteration. But they converges after 30th iteration which is the number of iterations required.

Comparing all the rankings generated from gibbs sampler and message passing, they are found to be not identical. The skill means estimated by the algorithms are different. However, Both algorithms usually assign the same ranks to the strong players, especially for the top 10 players.

The results of the gibbs sampler and messaging passing algorithm should be independent of initialization and pseudo random number seeds if the gibbs sampler has infinite time to sample. But the prior skill variance is found to affect the results.

The probailisties that the skill of one player is higher than the other are , and the probability of one player winning a match between the two is .

The difference between them is the noise () introduced in to the latter probability to account for performance inconsistency.

Table 4 favors players with lower skill means, and gives them higher probability of being more skilful than the opponent with higher skill means. Table 3 favors stronger players, and gives them higher probability to win.

Assume that the marginal skills are defined as,

Using the equation that compute Table 3 from part c, the probability that the player 1 has a better skill than player 16,

The probability that the player 16 has a better skill than player 1,



From 1100 iterations of skills samples from players 1 and 16, the first 4 samples are thrown away for the purpose of burn in time, and every 5th sample is taken because of the mixing time. What left are the 220 effective samples for each player. Then the number of times that player 1 has greater skill than player 16 is counted. This infers the probability.

These two alternatives describe the same distribution of gibbs samples. The first one is taking all the 220 gibbs samples for each player and computing out the mean and variance. Then put it into the equation to find the probability that one player has a higher skill. The second approach just infers the probability from the gibbs samples. If we have more gibbs samples, the second approach must converge to the the first approach. Therefore, the first approach is better.

Using the first method, the following table is computed.

Comparing Table 3 with Table 6, we see a similar result, maybe because message passing algorithem is just an approximation of MCMC gibbs sampling algorithm, and Gibbs sampler is not running long enough. Table 3 seems to give stronger player higher probability that he has a higher skill. But for players with similar skills, such as Nadal-Rafael and Federer-Roger, Table 6 assigns higher probability that a player has a higher skill to the stronger player.

Ranking based on empirical game outcome averages (ranged from -1 to 1), shown in Figure 9, shows more negative skills than other ranking systems. It is unfair to the player who played more and lost more and cannot really distinguish the players at the bottom. Since the range is smaller, it cannot rank the players distinctively.

Both rankings generated by gibbs sampling and message passing algorithms look similar and use full skill range from -1 to 2. But gibbs sampler has more positive skill means players than the message passing.