4F13 Coursework #2: Probabilistic Ranking

CCN: **5636B** 22nd Nov 2019 word count: 999

a Infer skills from Gibbs Sampling

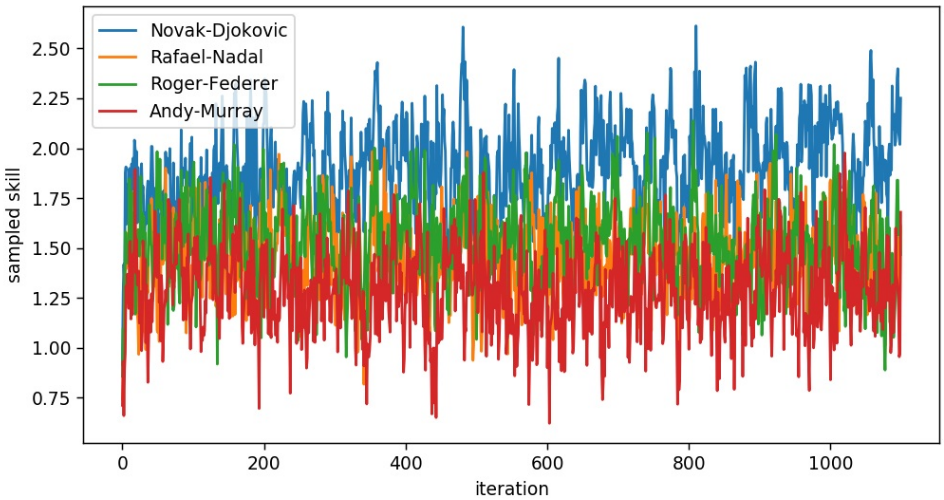


Figure 1: Gibbs - skill samples of the top 4 ATP players in the first 1100 iterations (30000 were run).

Gibbs sampling is an algorithm to implement Markov Chain Monte Carlo. When estimating a desired probability distribution from samples, two problems need to be addressed:

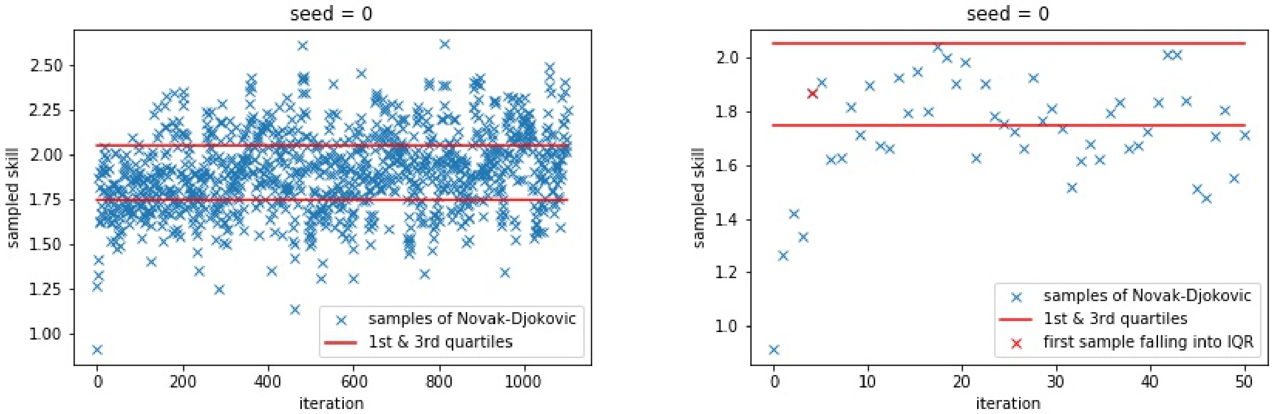
1. The skill value initialised for each player is likely to be different from their true skill. The samples are only valid after the Markov Chain has previously reached a **state of high probability**. The number of iterations necessary for **all player**'s samples to have fallen into their corresponding ranges of high probability is known as the **burn-in time**. Figure 2 demonstrates a reasonable way of determining it. The result is a very short **8 iterations**. Samples in this time range should be discarded in order to accurately estimate the desired skill distributions.

2. A sample from a Markov chain depends on the previous sample(s), hence consecutive samples are correlated. **Auto-correlation time** indicates how many iterations (lags) in between two samples will eliminate their correlation. Figure 3 shows that the longest auto-correlation time among all the players' samples is 14. For numerical simplicity, every 15th sample is kept in order to obtain uncorrelated samples.

Having solved these two problems, the algorithm should be run long enough for reliable results - stop when the histograms of samples have become fairly smooth (Figure 4).

```
games_won = np.where(G[:, 0] == p)[0]
games_lost = np.where(G[:, 1] == p)[0]
m[p] = np.sum(t[games_won]) - np.sum(t[games_lost])
...
iS[G[g,0], G[g,0]] += 1
iS[G[g,1], G[g,1]] += 1
iS[G[g,0], G[g,1]] += -1
iS[G[g,1], G[g,0]] += -1
```

Listing 1: Code for updating $\widetilde{\mu}$ and $\widetilde{\Sigma}$ in lecture slide.



(a) (b)

Figure 2: Determine **burn-in time** for Novak - given that the majority of samples of each player has shown a stationary distribution, as in Figure 1, the range of high probability can be estimated by the **inter-quartile range (IQR)** of the samples. (a) shows the samples the IQR of which is obtained, while (b) zooms into the first 50 iterations and identifies the first sample falling into the IQR. The overall burn-in-time is the maximum over all the players.

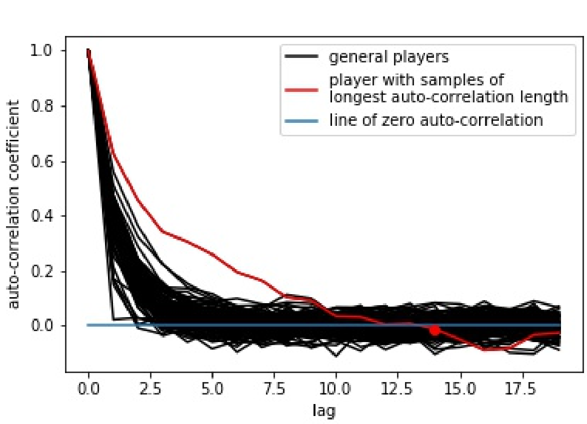


Figure 3: Variation of auto-correlation against lag for each player.

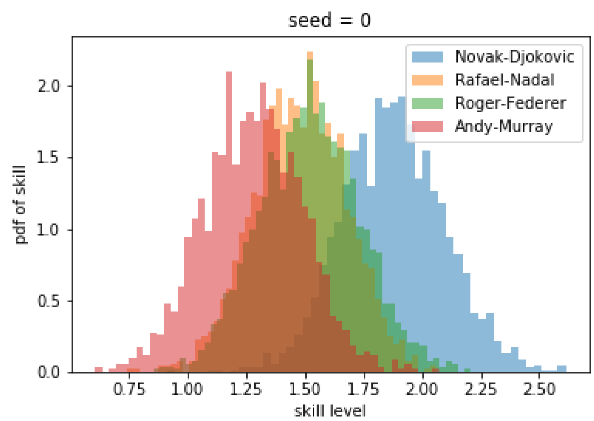
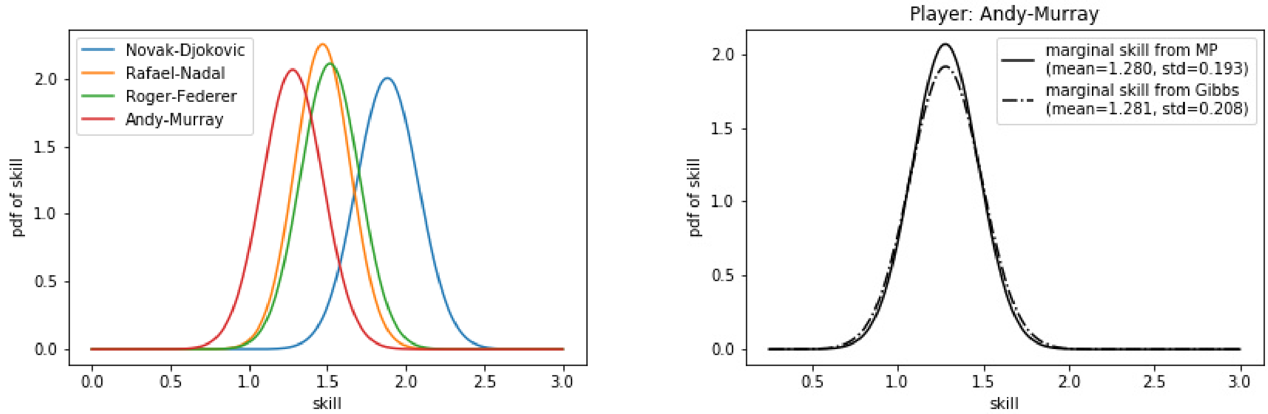


Figure 4: Histogram of samples (30000 iterations, thinned every 15th sample).

b Infer skills from Message Passing & Judge Convergences

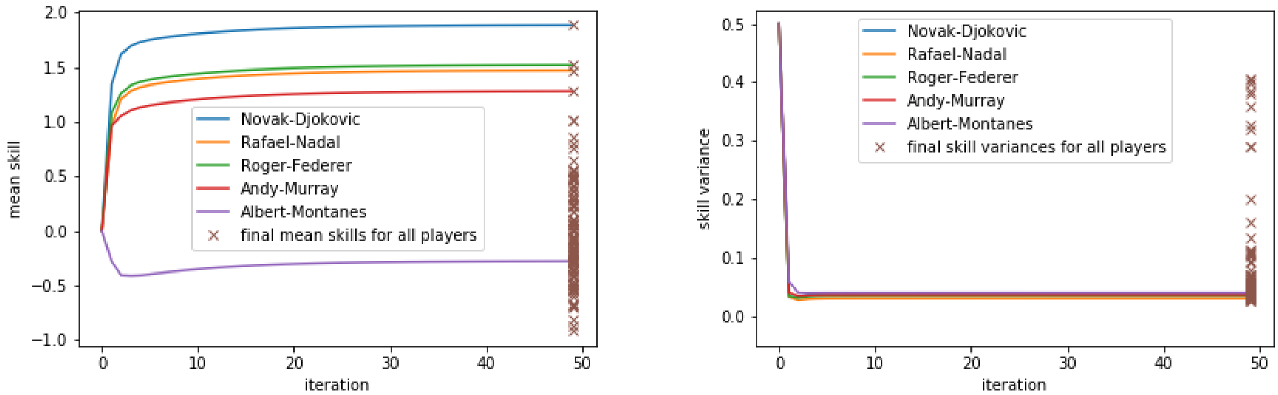
Figure 5a illustrates some results from Message Passing (MP) algorithm, whose means and variances match very well with those approximated from Gibbs samples (Figure 5b). The inference accuracy of both Gibbs and MP depends on what extent their results have converged to their corresponding objects (**bold text** below):

- Gibbs: The smoothness of the sample histograms mentioned in Figure 4 determines how much the samples have converged to **those which would have been generated by the desired distribution**. 30000 iterations was necessary.
- MP: The inferred mean and variance for each player's skill should converged to **those of the true skill**. Convergence is obtained when the inferred quantities no longer change as the iteration carries on (Figure 6). 50 iterations was necessary.



(a) (b)

Figure 5: MP - (a) Marginal skill Gaussians of the top 4 ATP players, and (b) its comparison with the marginal Gaussian approximated by Gibbs samples.



(a) (b)

Figure 6: Convergence of MP algorithm - (a) inferred mean, (b) inferred variance

```
mean_apx = np.mean(skill_samples_thinned[p,:])
var_apx = np.var(skill_samples_thinned[p,:])
plt.plot(x, norm.pdf(x, mean_apx, np.sqrt(var_apx)), 'k-.')
```

Listing 2: Code for approximating a Gaussian from Gibbs samples (Figure 5b).

c MP - probabilities of Higher Skill vs. Winning

In our generative model for a game outcome, noise of $\mathcal{N}(0,1)$ is added to the skill difference between two players ($s = w_1 - w_2$) to model their **performance inconsistency**. Hence the pdf of player 1's performance t will have a larger variance than that of s. The introduction of performance inconsistency gives rise to a discrepancy between the probabilities **that player 1 has higher skill** and **that player 1 will win**. Table 1 and 2 show the corresponding probabilities if the top 4 ATP players play against each other. It is observed that the probabilities of winning is closer to 0.5 than those of higher skill, due to performance inconsistency.

On a technical note, Figure 7 takes Djokovic vs. Nadal as an example and demonstrates how the probabilities of higher skill and winning can be computed as the area under Gaussians. The following code performs the calculations.

	Novak-Djokovic	Rafael-Nadal	Roger-Federer	Andy-Murray
Novak-Djokovic	-	**0.940**	0.909	0.985
Rafael-Nadal	0.060	-	0.427	0.766
Roger-Federer	0.091	0.573	-	0.811
Andy-Murray	0.015	0.234	0.189	-

Table 1: **MP** - Probability of one player (row label) has **higher skill** than another player (column label) among the top 4 players in ATP ranking.

	Novak-Djokovic	Rafael-Nadal	Roger-Federer	Andy-Murray
Novak-Djokovic	-	**0.655**	0.638	0.720
Rafael-Nadal	0.345	-	0.482	0.573
Roger-Federer	0.362	0.518	-	0.591
Andy-Murray	0.280	0.427	0.409	-

Table 2: **MP** - Probability of one player (row label) will **win** another player (column label) in a future match among the top 4 players in ATP ranking.

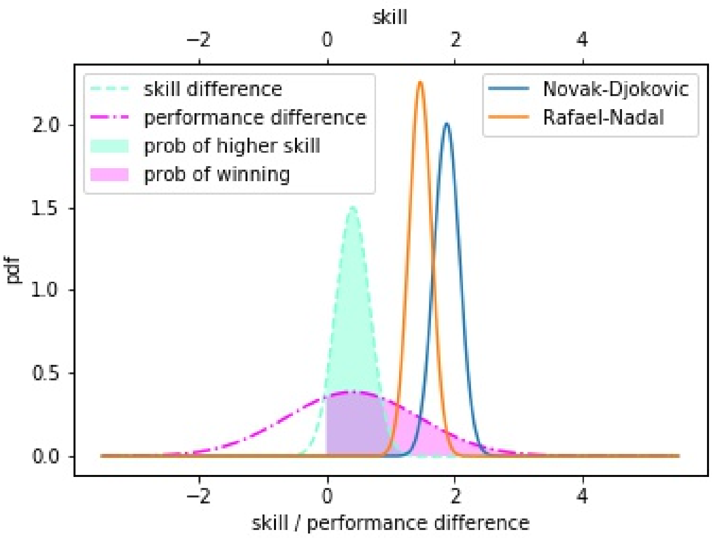


Figure 7

```
mean_s = mean_p1 - mean_p2
var_s = var_p1 + var_p2
prob_higher_skill = 1 - norm.cdf(0, mean_s, np.sqrt(var_s))
prob_winning = 1 - norm.cdf(0, mean_s + 0, np.sqrt(var_s + 1))
```

Listing 3: Code for computing the probabilities of higher skill and winning.

d Gibbs - 3 ways of obtaining probabilities of Higher Skill

Given the Gibbs samples, there are 3 different ways of calculating the probability of higher skill, each of which is explained below:

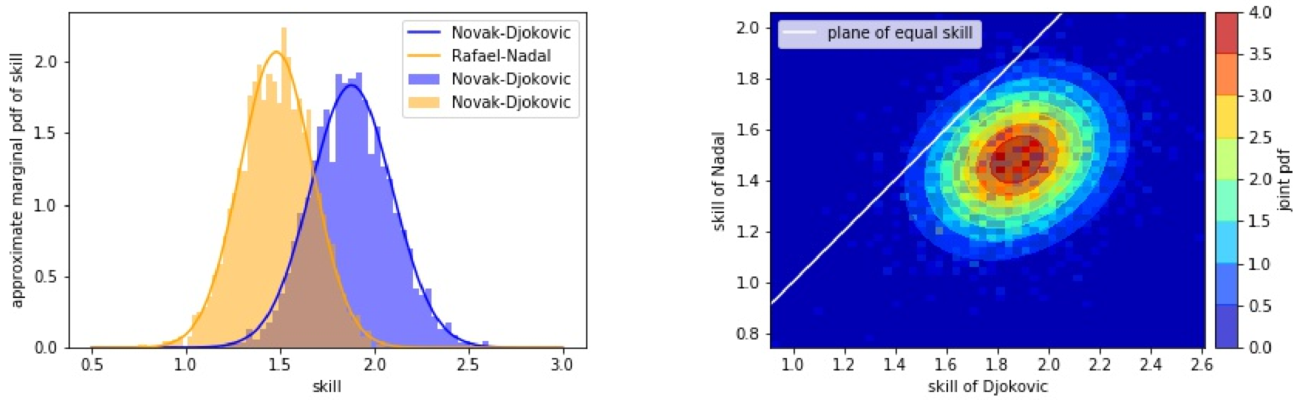
- **From two approximated marginal Gaussians**: A marginal Gaussian for each player's skill can be constructed by calculating the mean and variance of all the samples of that player.

Then with marginal Gaussians for two players (Figure 8a), the same procedures in Section (c) can be followed to obtain the probability of higher skill.

- **From one approximated joint Gaussian**: Treating the samples of two players at each iteration as a 2D datapoint, a 2D Gaussian can be estimated instead (Figure 8b). The probability of higher skill is simply the volume **underneath this Gaussian pdf** and **to the right of the "vertical plane of equal skill"**.

- **Directly from samples**: Compare samples of two players at each iteration and take the proportion that player 1's sampled skill is higher than player 2's as the probability of higher skill.

$$P(w_1 > w_2) = \frac{\text{number of iterations } i \text{ where } w_1^i > w_2^i}{\text{total number of iterations}}$$



(a) (b)

Figure 8: Approximate (a) marginal and (b) joint Gaussian(s) from Gibbs samples.

Table 3 shows the corresponding results. A simple analysis for the discrepancies is that the two separate marginal Gaussians fail to capture the correlation between the two player's skills which the single joint Gaussian can. However the joint Gaussian method actually tries to achieve the same calculation as the direct method but has intermediate steps. (The direct method can be seen as finding the proportion of samples lying on the right of the plane of equal skill in Figure 8b.) Hence direct method is the best because, with the same samples, the intermediate steps of joint Gaussian method cannot increase the prediction accuracy.

Method	Marginal Gaussians	Joint Gaussian	Directly from samples
Probability	0.915	0.941	0.950

Table 3: Probabilities that Djokovic has higher skill than Nadal according to the three methods.

Using the direct method, Gibbs produces a table of higher skill probabilities (Table 4) very similar to that of MP. The largest difference and % difference of the probabilities are only +0.025 (Rafael-Nadal vs. Andy-Murray) and +5.4% (Rafael-Nadal vs. Roger-Federer) respectively.

```
joint_samples = np.stack((p1_thinned_samples, p2_thinned_samples), axis=0)
mean_vector = np.mean(joint_samples, axis = 1)
cov_matrix = np.cov(joint_samples)
```

Listing 4: Code for approximating a joint Gaussian between two players' skills.

	Novak-Djokovic	Rafael-Nadal	Roger-Federer	Andy-Murray
Novak-Djokovic	-	0.950	0.913	0.993
Rafael-Nadal	0.050	-	0.450	0.791
Roger-Federer	0.083	0.550	-	0.804
Andy-Murray	0.007	0.209	0.196	-

Table 4: **Gibbs** - Probability of one player (row index) has **higher skill** than another player (column index) among the top 4 players in ATP ranking.

e Empirical & Probabilistic Rankings

In this last section, players' rankings are produced by three different systems (Figure 9).
A general player i's ranking scores are computed as follows:

(a) **Empirical Ranking:**

$$\text{proportion of wins} = \frac{\text{number of wins}}{\text{total number of games played by } i}$$

(b) **Probabilistic Ranking via Gibbs Sampling:**

$$\text{average probability of winning} = \frac{1}{106} \sum_{j \neq i} P(\text{player } i \text{ wins } j)$$

where $P(\bullet)$ is the probability of winning inferred from the marginal skill Gaussians of the two players. Marginals are approximated from Gibbs samples.

(c) **Probabilistic Ranking via Message Passing:**

Same ranking criterion as (b). However in MP algorithm the marginals are derived exactly.

Comparing the three ranking lists has led to the following interpretations:

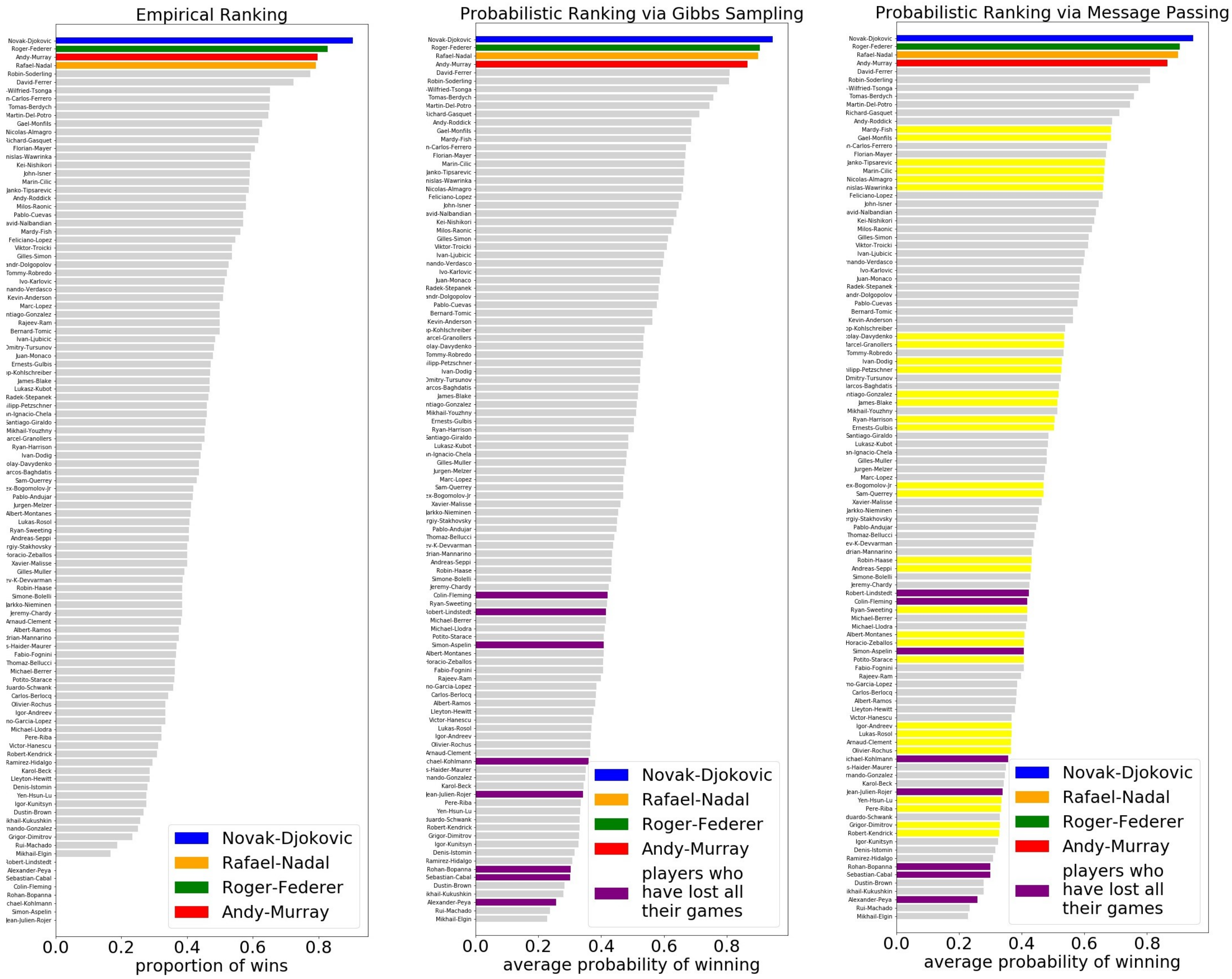
1. The top 4 ATP players have different ranking orders in Empirical system and Probabilistic systems - the positions of Rafael-Nadal and Andy-Murray are swapped. This observation can be explained by the fact that although the proportion of games Murray won is slightly higher than Nadal, the Empirical system doesn't take into account who they have played against, i.e. how skilled their opponents were. Both Probabilistic systems conclude that Nadal's opponents are more skilled hence he should deserve higher rank.

2. Players who won no games are ranked at the bottom in Empirical system, which again does not consider how skillful their opponents were. Probabilistic systems infer their skill level from who their opponents have also played against. Hence their ranks move up (purple bars in Figure 9b and 9c).

3. Comparing between Gibbs and MP rankings, the yellow bars in Figure 9c are the players whose ranks are different in the two systems. These local ranking differences is due to the small discrepancies between the Gibbs and MP marginals, as demonstrated in Figure 5b.

```
probs_win_p1 = []
for p2 in range(M):
    if p2 != p1:
        ...
        probs_win_p1.append(prob_win)
avg_prob_win_p1 = sum(probs_win_p1) / (M-1)
```

Listing 5: Code template for computing a player's probablistic ranking score for both Gibbs and MP.



(a) (b) (c)

Figure 9: Three different ranking systems.