4F13 - COURSEWORK #2: PROBABILISTIC RANKING

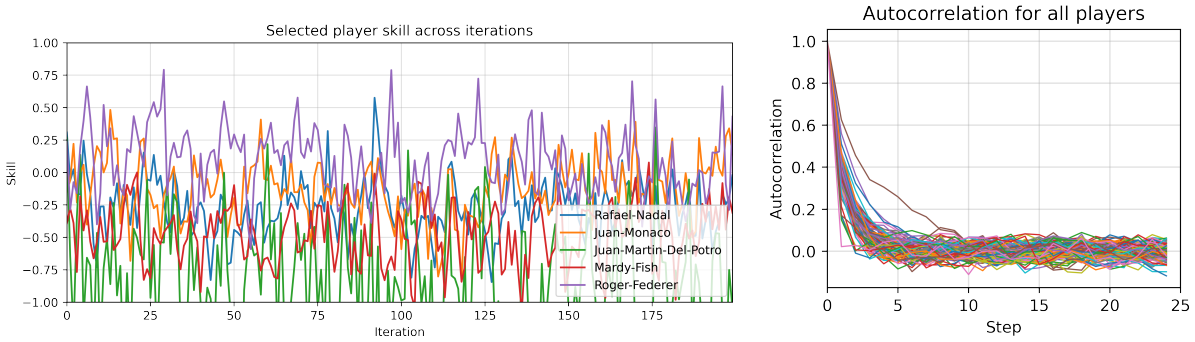
CCN: 5654F

1. Task A: Gibbs Sampling

The code in `gibbsrank.py` was completed by adding functionality to generate the sample skills given performance differences by adding line 1 of Listing 1 to the first uncompleted loop over all the players (which negates the skill difference of lost games from those won). Lines 4 onwards of the same Listing were then added to build the `iS` matrix, by adding ± 1 to each entry to match the form of the covariance matrix in the lecture notes[1].

```
1           m[p] = sum(t[np.where(G[:, 0] == p)]) - sum(
2                   t[np.where(G[:, 1] == p)]
3               )
4
5           winner = G[g, 0]
6           loser = G[g, 1]
7
8           iS[winner, winner] += 1
9           iS[loser, loser] += 1
10          iS[winner, loser] -= 1
11          iS[loser, winner] -= 1
```

Listing 1. Task A code excerpts



(a) Player Skills

(b) Player Autocorrelations

Figure 1. Task A Figures

The Gibbs sampler was then run for 1100 iterations using the provided python notebook, and 4 random players skills were plotted across the first 200 iterations: Figure 1a. The data is noisy but is clearly not completely independent as each skill point stays fairly close to the previous point. The autocorrelation for each player for 25 steps is plotted in Figure 1b, which shows that an autocorrelation time of at least 10 samples is needed for the autocorrelation over the skills to be 0 for all athletes. Plotting the average population mean and variance also shows that the burn-in time (time for the chain to converge to the distribution) is lower than the 10 samples needed for the samples to become independent. Therefore the first 10 samples need to be discarded.

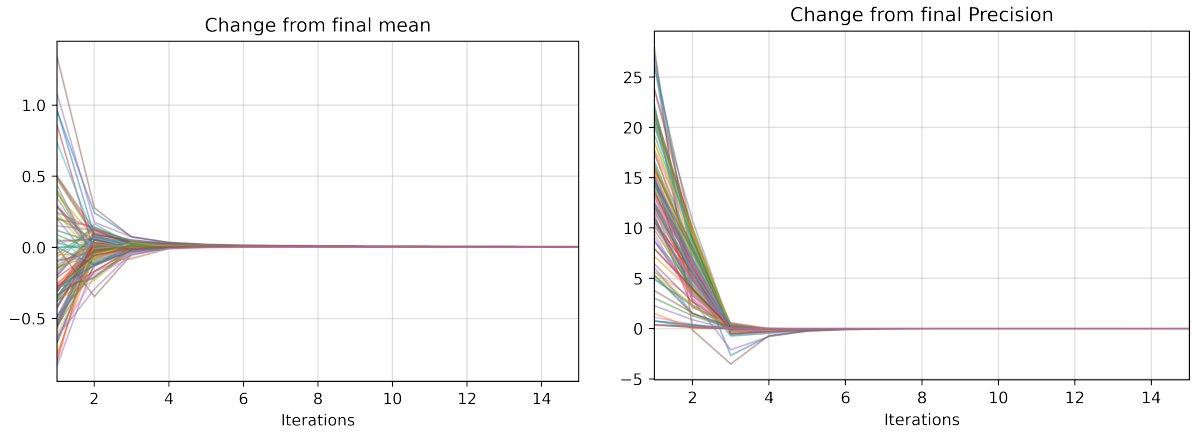
2. Task B: Message Passing

Convergence occurs when the parameters *converge* to a constant value (the true value according to the algorithm). For Gibbs Sampling this is when the the Markov chain converges to a stationary probability distribution (the joint skill distribution). For message passing/EP this

Date: Wednesday 15th November 2023.

[1]CUED 4F13 Lecture notes (course site)

occurs when a stable graph is reached, each iteration of messages passed does not affect the mean and precision of each vertex. In Section 1 we saw that the `max`(burn-in time, autocorrelation time) was ∼10 samples. A new function was first added to the `eprank.py` that returns the means and precisions at each iteration, which allowed graphs of the change in mean and precision from their final values to be produced as Figure 2. These show a more rapid convergence than the Gibbs sampling method, with the player means and precisions each reaching their final values in only ∼6 iterations, a 40% improvement over the method in Section 1.



(A) Player Mean Convergence (B) Player Precision Convergence

FIGURE 2. Task B Figures

3. TASK C: PLAYER RANKING USING EP

If players i, j have skills w_i, w_j, $p(w_i > w_j)$ can be easily calculated using the definitions in the lecture notes[2] as in Equation 2. Similarly, a player wins against another in this model if the skill difference plus a noise $n \sim \mathcal{N}(0, 1)$ is greater than zero for i, or less than zero for j. Therefore the probability of i winning is given by Equation 2, where the λs are the precisions.

$$p(w_i > w_j) = p(w_j - w_i < 0) = \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\lambda_i^{-1} + \lambda_j^{-1}}}\right) \tag{1}$$

$$p(w_i - w_j + n > 0) = \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\lambda_i^{-1} + \lambda_j^{-1} + 1}}\right) \tag{2}$$

```
1        prob_has_better_skill = 1.0 - norm.cdf(
2            0, mean_differences, vars_sums**0.5
3        )
4
5        prob_win = 1.0 - norm.cdf(
6            0, mean_differences, (vars_sums + 1.0) ** 0.5
7        )
```

LISTING 2. Task C code excerpts

Lines 1 and 4 of Listing 2 are therefore used to calculate a matrix showing the probabilities of the top 4 ranked players (from the lecture notes' ranking[3]) a. having a higher skill than the others, and b. their probability of winning if playing against the other players. These lines

[2] see footnote 1
[3] see footnote 1

—	D	N	F	M
Djokovic(D)	0.500	0.940	0.909	0.985
Nadal(N)	0.060	0.500	0.427	0.767
Federer(F)	0.091	0.573	0.500	0.811
Murray(M)	0.015	0.233	0.189	0.500

TABLE 1. p(Higher Skill)

—	D	N	F	M
Djokovic(D)	0.500	0.655	0.638	0.720
Nadal(N)	0.345	0.500	0.482	0.573
Federer(F)	0.362	0.518	0.500	0.591
Murray(M)	0.280	0.427	0.409	0.500

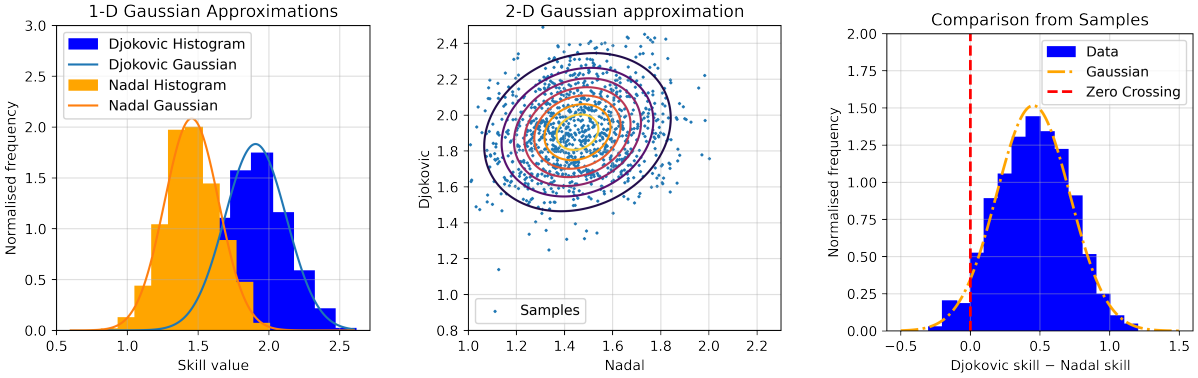
TABLE 2. p(Win)

just demonstrate the equations from the previous paragraph, using the `scipy.stats.norm.cdf` function for Φ. The results are shown in Tables 1 and 2 respectively.

It can clearly be seen that a player with higher expected skill than another is always expected to win the match between them, and the converse. However, the probability of winning is much lower than the probability of having a higher skill due to the additional match variance term n (the performance noise). This adds to the model the fact that a player can have an "off match", and aids in training by making the model less confident in attributing skill to the winning player of each match.

4. Task D: Nadal and Federer Skill Comparison

4.1. **Approximation of Marginal Skills by Gaussians.** The parameters of each Gaussian were calculated by taking the mean and variance of the Gibbs samples after training (excluding burn-in/autocorrelation time) for each player. These are plotted in a histogram to justify the use of a Gaussian (which we expect to be the case due to the properties of the model), with the Gaussian fit alongside. This is shown in Figure 3a. As expected - as he is ranked higher - Djokovic's mean skill is higher than Nadal's (1.899 vs 1.463), although the variance in the skill is higher (0.224 vs 0.188).



(A) Gaussian Marginal Skill (B) Gaussian Joint Skill (C) Direct Comparison

FIGURE 3. Task D Figures

4.2. **Approximation of Joint Skills by Gaussian.** The same analysis is done, but by fitting a joint gaussian to the burned-in Gibbs samples by using `numpy` to generate the covariance matrix and the means. The means are unsuprisingly the same, and the covariance matrix is shown in Equation 3. A contour plot of the 2-D Gaussian Generated is shown in Figure 3b. The covariance matrix shows a small diagonal term (~ 0.074) showing that the two players skills are slightly correlated.

$$\Sigma = \begin{bmatrix} 0.0367 & 0.0074 \\ 0.0074 & 0.0474 \end{bmatrix} \tag{3}$$

4.3. **Direct Skill Comparison from samples.** In this method the burned-in skill samples are simply negated from each other, and a histogram drawn and a gaussian trained on those differences. The resulting plot is shown in Figure 3c.

—	D	N	F	M
Djokovic(D)	0.500	0.957	0.922	0.982
Nadal(N)	0.043	0.500	0.377	0.736
Federer(F)	0.078	0.623	0.500	0.801
Murray(M)	0.018	0.264	0.199	0.500

TABLE 3. Skill Probabilities

—	D	N	F	M
Djokovic(D)	0.000	0.017	0.014	-0.003
Nadal(N)	-0.017	0.000	-0.050	-0.030
Federer(F)	-0.014	0.050	0.000	-0.010
Murray(M)	0.003	0.030	0.010	0.000

TABLE 4. Difference to Table 1

4.4. **Skill comparison.** As there is a non-zero diagonal term in the covariance matrix in method 2, and a visible correlation in the figure, method 2 is preferred over method 1. The direct method also requires a lot of samples to avoid significant noise in the probabilties. For a real life task, therefore, when computation time is at a premium, the best is method two.

The skill probabilities are therefore calculated in a very similar manner to that in Section 3, but the formula must be changed to include the diagonal covariance terms, leading to Equation 4. A similar table is then generated as seen in Table 3, with the diagonals set to 0.5. This is very close to the table produced in Section 3, and the (very small) differences are shown in Table 4. Both methods produce very similar results and are therefore producing very similar distributions.

$$p(w_i > w_j) = p(w_j - w_i < 0) = \Phi\left(\frac{\mu_i - \mu_j}{\sigma_i^2 - 2\sigma_i j^2 + \sigma_j^2}\right) \tag{4}$$

5. Task E: Player Ranking Comparisons

5.1. **Empirical Game Outcome Averages.** The ranking is found by finding the total number of games won by each player, and ranking the players by their win ratio, which has the underlying assumption that the probability of winning a game is the same regardless of the opponent played against.

5.2. **Gibbs Sampling.** After the gibbs sampling exercises from previous tasks it is easy to empirically calculate the mean and variance for each player, as was done when calculating the marginals in Section 4.

5.3. **Message Passing Algorithm.** The algorithm is run for enough iterations to converge (≥ 6) which returns the means and precisions directly.

The mean and $\pm\sigma$ error bars for the probability of winning against a randomly chosen player for all three methods are shown in Figure 4, ranked by the message passing algorithm generated means. There are a few obvious issues with the empirical game outcome average results. Firstly the range in the means is a lot lower than the other algorithms, further reinforcing the point from section 4 that a far greater amount of data than we have is needed to ensure convergence. More importantly, players cannot be compared if they have the same win ratio. This is clearest for players who have won no games. Intuitively, players who have lost against harder players are maybe more likely to be better than players who have lost against players who are proven to be bad from other results. This explains why there are players with a win ratio of 0 who are in the middle of the ranking when ranked by the EP process. The win ratio method does not take into account anything about the skill of opponents.

The EP and Gibbs results give broadly similar results. The shape of the means is the same, the means of the two processes seem separated by a constant over all players, but the ranking overall is very similar. The constant arises from the prior over the skills. The variances in EP and Gibbs also match closely, showing again that the two methods result in similar distributions. The players with a win ratio of 0 have a larger variance in both the Gibbs and EP processes also, reinforcing the lack of information gained about their skill.

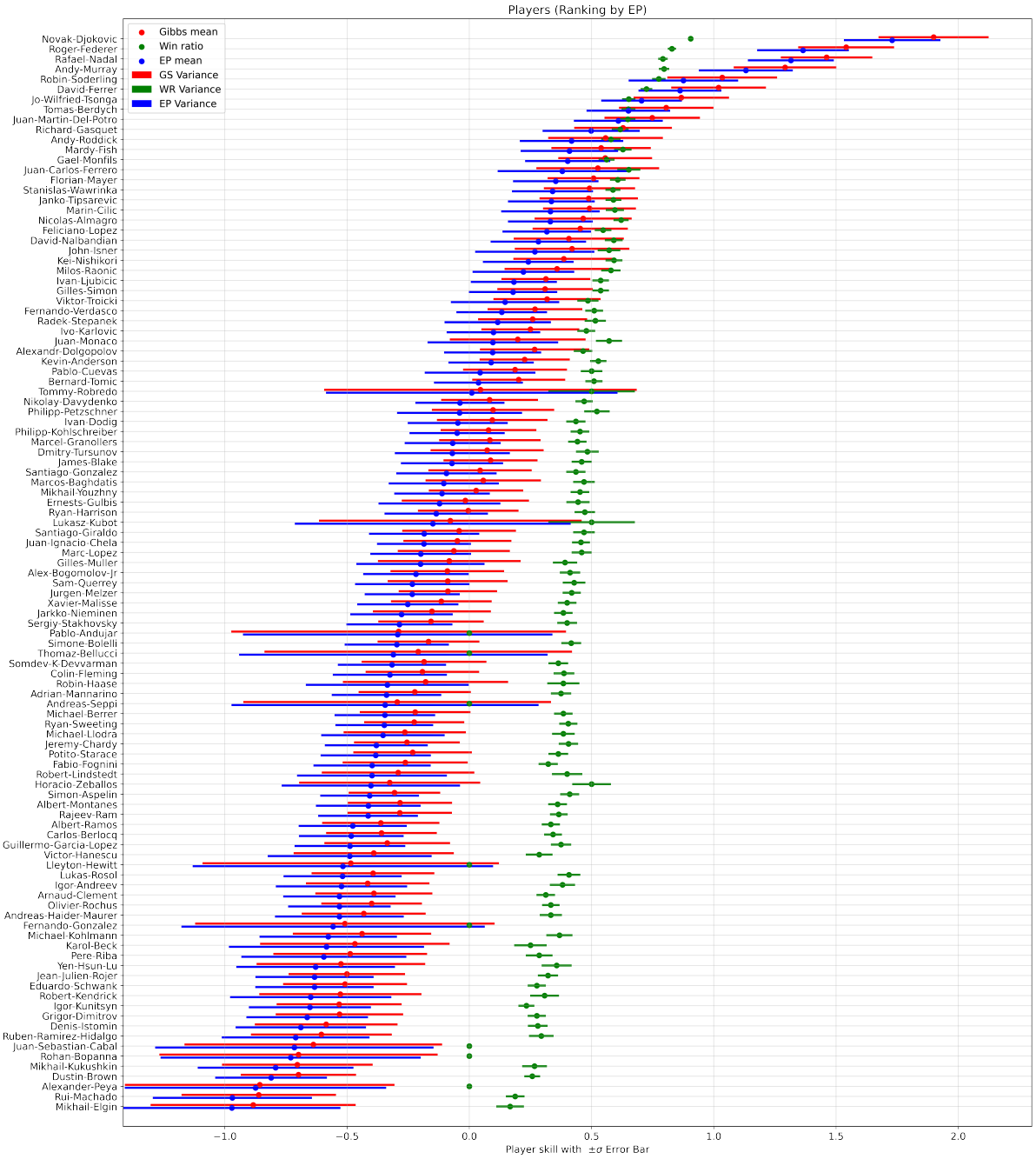


FIGURE 4. Player Rankings