4F13 Probabilistic Machine Learning
Coursework 2: Probabilistic Ranking

Candidate Number: 5628A

Word count: 1000

Section a

The Gibbs sampler is run for 10000 iterations and some sample sequences are plotted (Figure 1).

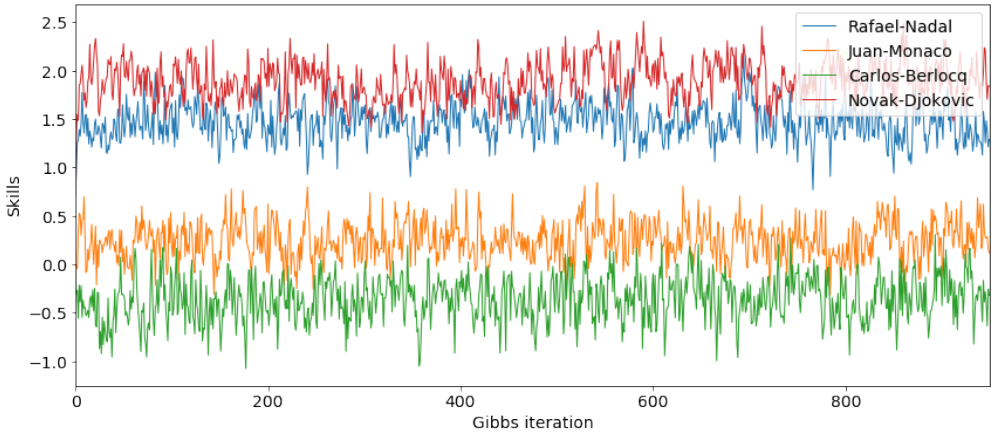


Figure 1: Example Gibbs sequences of players' skills (first 950 samples)

Listing 1: Code for computing the posterior mean and likelihood precision in `gibbsrank.py`

```
m = np.zeros((M, 1))
for p in range(M):
    tmp = (G[:, 0] == p).astype(np.int) - (G[:, 1] == p).astype(np.int)
    m[p] = np.dot(t.flatten(), tmp)

for g in range(N):
    ig, jg = G[g]
    iS[ig, ig] += 1
    iS[jg, jg] += 1
    iS[ig, jg] -= 1
    iS[jg, ig] -= 1
```

Burn-in time

Since the sampler is initialised randomly according to the prior distribution, it is likely that the initial samples are in the low-density region of the desired distribution. Figure 2 shows that most of the samples have entered the high-density region after 5 iterations but to be safe, we take the burn-in time to be 20 iterations.

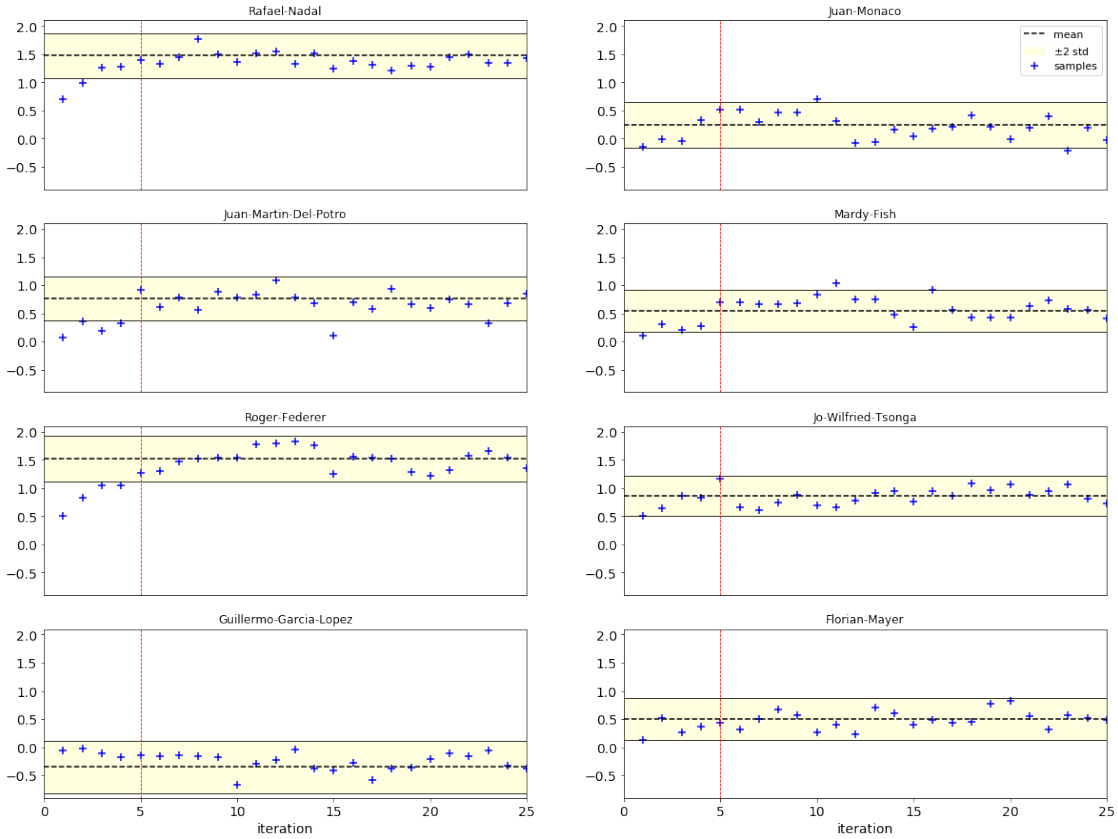
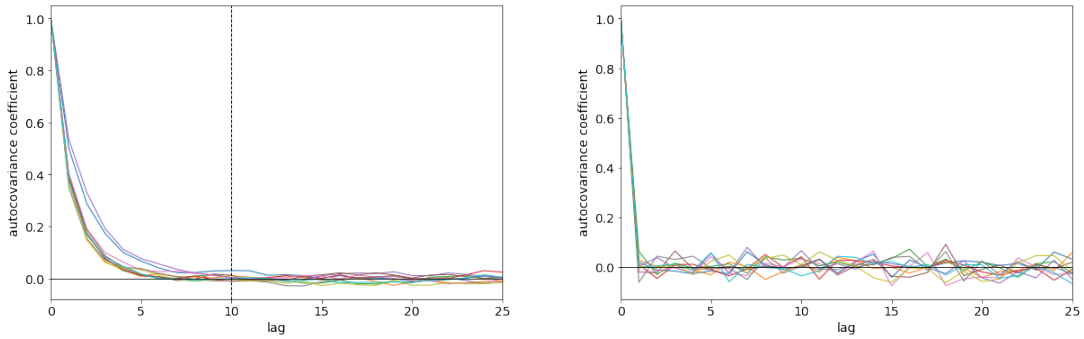


Figure 2: First 25 Gibbs skill samples of 8 random players. For del Potro and Federer, the Gibbs sampler takes 4 iterations to enter the high density region (± 2 standard deviation).

Auto-correlation time

If subsequent samples have highly positive correlation, the sampler would need more iterations to "explore" the distribution. Thus, we want samples that are independent of or even negatively correlated to the previous samples for faster convergence and reduced storage. Figure 3 shows that at 10 iterations apart, the samples have close to zero correlation and hence, we "thin" the samples by a factor of 10 (Figure 4).



(a) (b)

Figure 3: Plots of auto-covariance coefficient against lag for 10 random players' skill sequences. Panel (a) shows the auto-covariance coefficients for the original Gibbs samples. Panel (b) shows the auto-covariance coefficients for the thinned samples (by a factor of 10); the correlation between subsequent samples are much closer to zero.

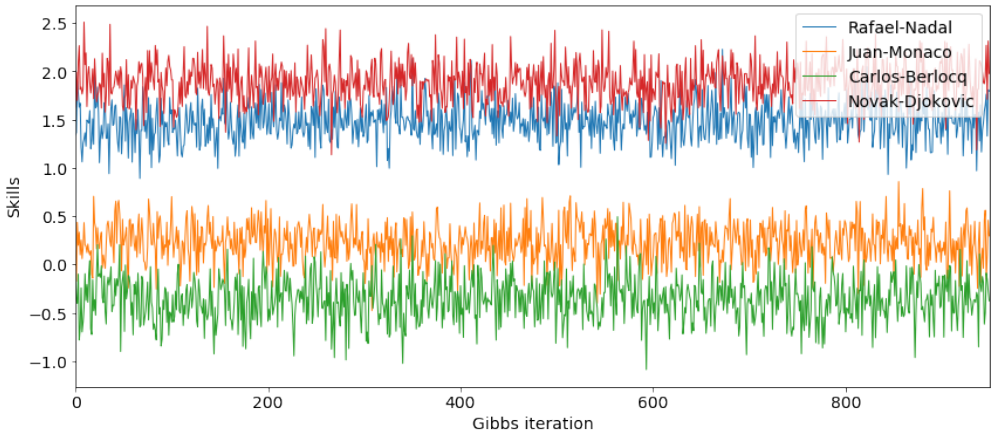


Figure 4: Example Gibbs sequences after burn-in and thinning. Compared to Figure 1, the thinned samples have bigger step sizes in the variable space and are able to explore the distribution in fewer iterations.

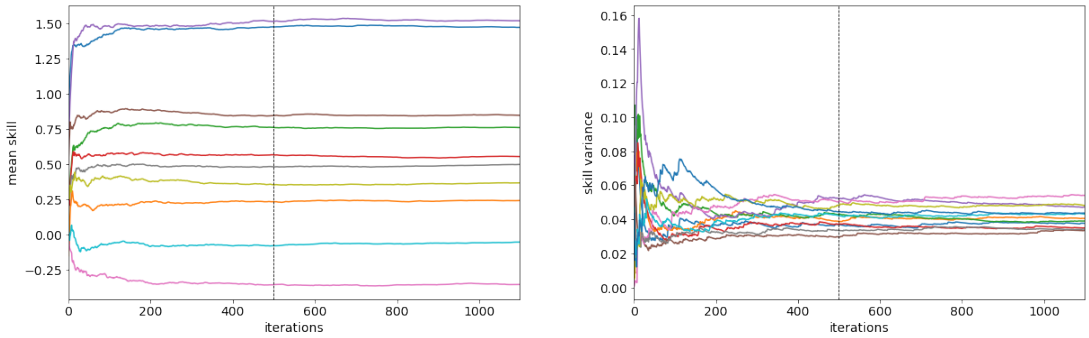
To obtain reliable results in calculations, around 1000 samples after burn-in and thinning are needed; thus, 10000 original samples are desired.

Section b

Convergence in Gibbs sampling

Convergence in Gibbs sampling means that the random variables $\mathbf{w}$, $\mathbf{t}$ are sampled from the conditional distributions, $p(\mathbf{w}|\mathbf{y},\mathbf{t})$ and $p(t_g|\mathbf{y},\mathbf{w})$, as if they were sampled from the true joint posterior distribution $p(\mathbf{w},\mathbf{t}|\mathbf{y})$. In other words, the Markov Chain has stabilised to its stationary distribution (joint posterior), starting from the priors.

One way to judge convergence is to observe whether the statistics of the samples have stabilised. Figure 5 shows that the mean and variance have stabilised after 500 iterations, indicating convergence.



(a) (b)

Figure 5: Plots showing the mean (Panel (a)) and variance (Panel (b)) of 10 random players' Gibbs samples against iteration. At iteration i, the statistics are calculated using samples from iteration 0 to iteration i. After iteration 500, the mean and variance are both steady, indicating the MC has converged.

A more formal way of assessing convergence is the *Geweke diagnostic* (Geweke 1992): the samples after mixing time are split into two sequences, the first 10% (*sequence 1*) and the last 50% (*sequence 2*). The *Geweke z-score* for each player's skill w_i is then computed (Figure 6):

$$\text{Geweke z-score} = \frac{\mu_{seq1} - \mu_{seq2}}{\sqrt{\sigma^2_{seq1} + \sigma^2_{seq2}}}$$

The standard errors are all within 0.05 std's and hence we can conclude that the MC has converged to its stationary distribution.

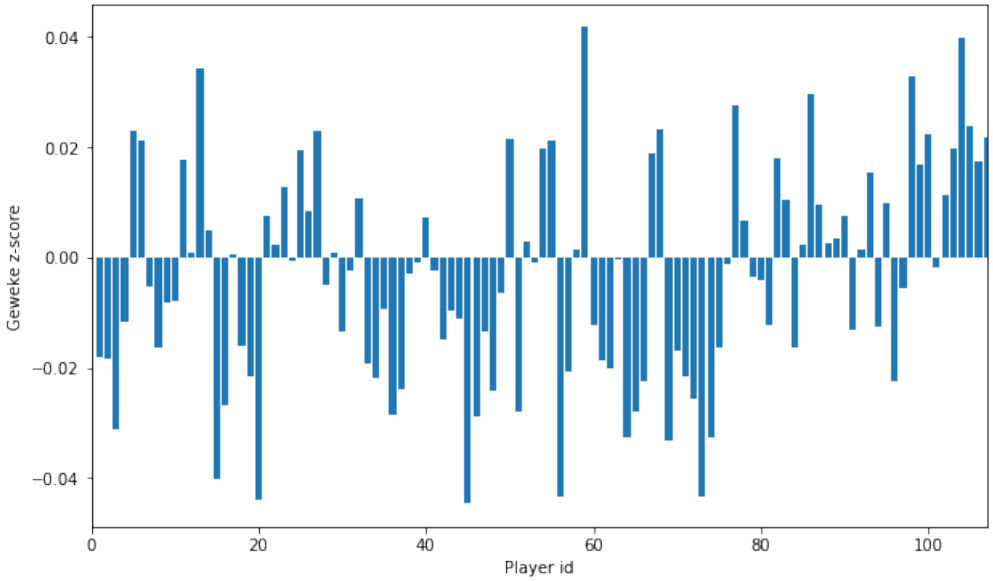
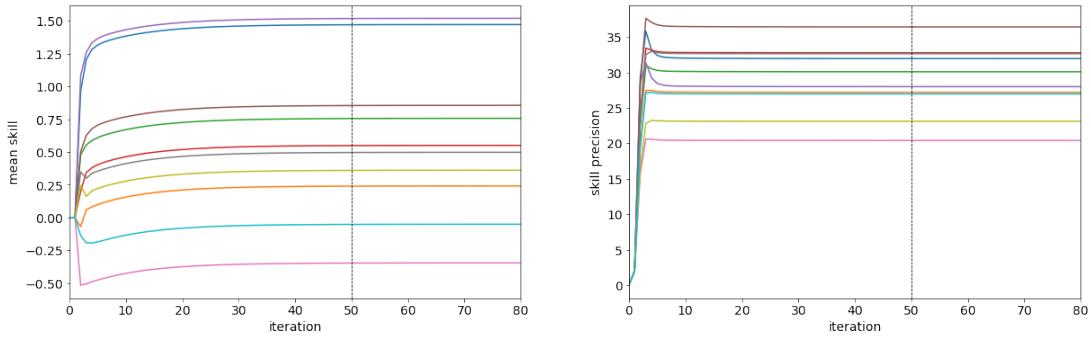


Figure 6: Geweke z-score for each of the player's skill sequence

Convergence in message passing

For message passing in TrueSkill, convergence means that the marginals of the variable nodes have converged to the exact/approximated marginals. In our case, the marginals ($\mathbf{w}$ and $\mathbf{t}$) are approximated by Gaussians and therefore we can judge convergence by checking whether the means and precisions of the skills have stabilised. Figure 7 shows that around 50 iterations are required for convergence.



(a) (b)

Figure 7: Plots of mean and precision of 10 random players' skills using the message passing algorithm, against the number of iteration

Section c

The probability that Player i has higher skill than Player j is:

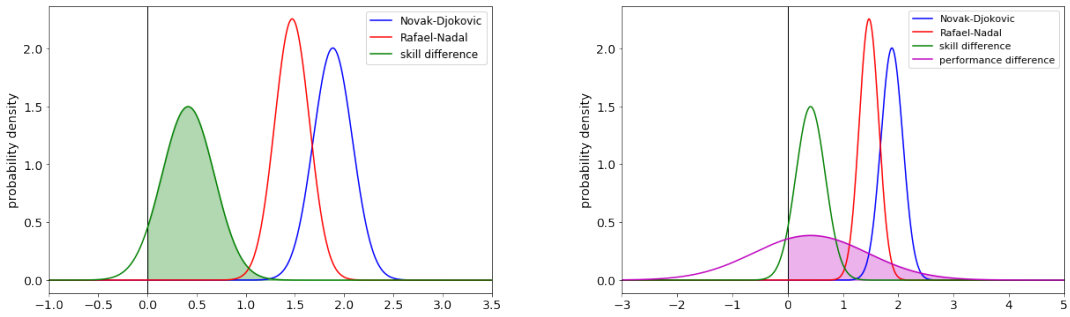
$$p(w_i > w_j) = \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right) \tag{1}$$

To compute the probability of Player i winning a match against Player j, Gaussian noise $n \sim \mathcal{N}(0, 1)$ is added to the skill difference:

$$\text{performance difference: } t = w_i - w_j + n$$

$$p(t > 0) = \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2 + 1}}\right) \tag{2}$$

These calculations are visualised in Figure 8. The 4×4 tables of probabilities are recorded in Table 1 and Table 2. The main difference between these two tables is that Table 2 accounts for performance inconsistency which adds uncertainty to the game outcome.



(a) (b)

Figure 8: Plots showing the marginals of Djokovic and Nadal's skills approximated using the message passing algorithm. Panel (a) shows the distribution of the skill difference (green curve); the probability that Djokovic has a higher skill than Nadal is the area shaded in green. Panel (b) shows the distribution of the performance difference (purple curve); the probability that Djokovic beats Nadal in a match is the area shaded in magenta.

	Djokovic	Nadal	Federer	Murray
Djokovic	-	0.949	0.909	0.985
Nadal	0.060	-	0.427	0.767
Federer	0.091	0.573	-	0.811
Murray	0.015	0.233	0.189	-

Table 1: Table of probability that *one player has higher skill than the other*. Entry at row i and column j represents the probability that Player i has higher skill than Player j.

	Djokovic	Nadal	Federer	Murray
Djokovic	-	0.655	0.638	0.720
Nadal	0.345	-	0.482	0.573
Federer	0.362	0.518	-	0.591
Murray	0.280	0.427	0.409	-

Table 2: Table of probability of *one player winning a match against the other*. Entry at row i and column j represents the probability of Player i winning a match against Player j.

Listing 2: Code for computing the probabilities in Table 1 and Table 2

```
def prob_w1_gt_w2(samples1, samples2):
    """Probability that player1 has higher skill than player2"""
    mu1, v1 = np.mean(samples1), np.var(samples1)
    mu2, v2 = np.mean(samples2), np.var(samples2)
    return norm.cdf((mu1 - mu2) / np.sqrt(v1 + v2))

def prob_p1_beats_p2(samples1, samples2):
    """Probability that player1 wins a match against player2"""
    mu1, v1 = np.mean(samples1), np.var(samples1)
    mu2, v2 = np.mean(samples2), np.var(samples2)
    return norm.cdf((mu1 - mu2) / np.sqrt(v1 + v2 + 1))
```

Section d

Using the Gibbs sequences, we compare the skills of Djokovic and Nadal by approximating $p(w_{Djokovic} > w_{Nadal} \mid \mathbf{y})$, using three different methods.

Gaussian approximation of marginal skills (method 1)

Each player's skill marginal is approximated by a Gaussian with mean and variance equal to those of the player's skill samples. $p(w_{Djokovic} > w_{Nadal} \mid \mathbf{y})$ can then be computed using eq. 1, which gives 92.4%.

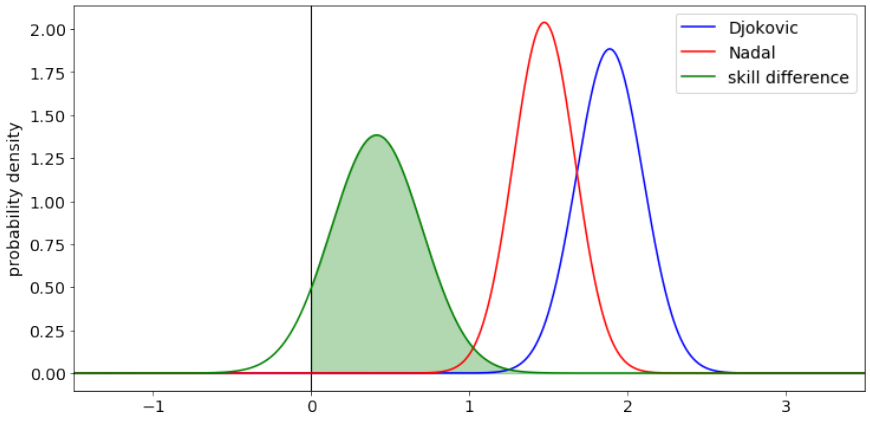


Figure 9: Marginals of Djokovic and Nadal's skills approximated as Gaussians using Gibbs samples. The green curve represents the distribution of the skill difference and the area shaded in green is the approximated probability that Djokovic has a higher skill than Nadal.

Gaussian approximation of joint skills (method 2)

The joint skills of the two players are approximated by a 2-d Gaussian with means and covariance matrix calculated using the two players' skill samples. $p(w_{Djokovic} > w_{Nadal} | \mathbf{y})$ is obtained by (see Appendix for derivation):

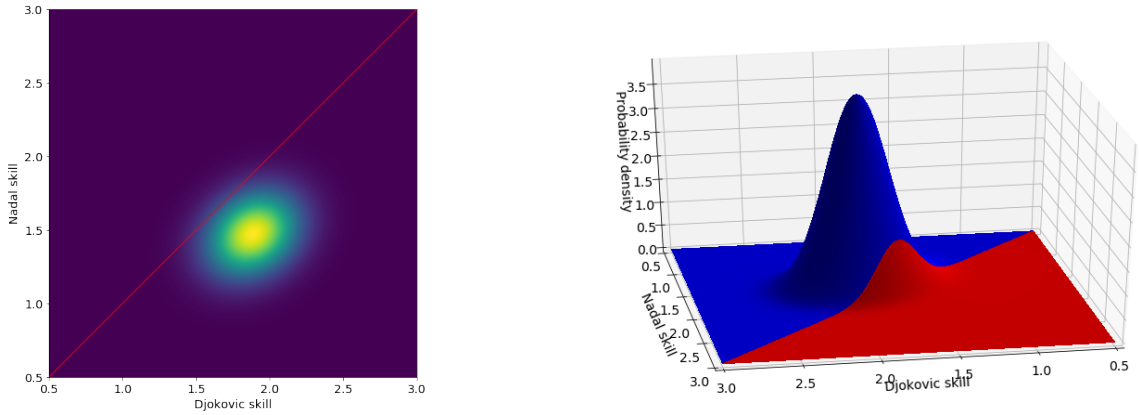
$$p(w_i > w_j) = \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2 - 2\sigma_{ij}}}\right), \tag{3}$$

where σ_{ij} is the covariance between sequence i and sequence j.

The approximated $p(w_{Djokovic} > w_{Nadal} \mid \mathbf{y})$ is 94.8%.

Direct approximation from samples (method 3)

To approximate $p(w_{Djokovic} > w_{Nadal})$, we calculate the fraction of the Gibbs samples where $w_{Djokovic} > w_{Nadal}$, which is 95.1%.



(a) 2-d heatmap

(b) 3-d plot

Figure 10: Joint distribution of Djokovic and Nadal's skills approximated by a 2-d Gaussian using Gibbs samples. Panel (a) and Panel (b) show the same distribution using different plot types. In Panel (a), we can observe the positive correlation. In Panel (b), the volume under the blue surface is the approximated probability that Djokovic has a higher skill than Nadal. Notice that the 3-d plot is rotated for better visualisation.

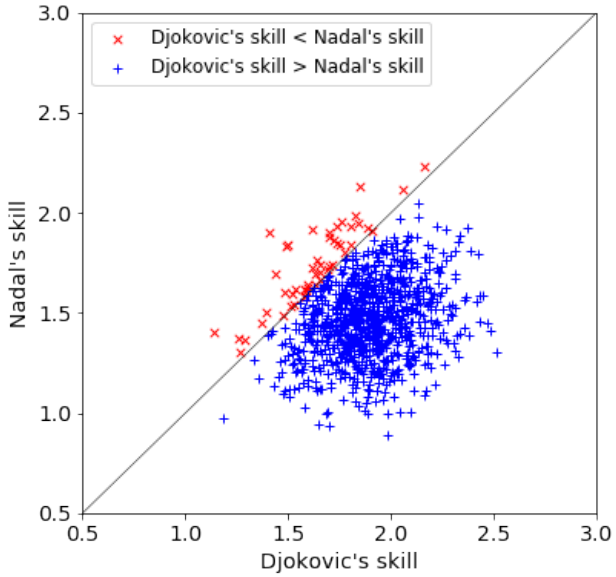


Figure 11: Scatter plot of Gibbs samples of Djokovic and Nadal's skills

Listing 3: Code for estimating the probability that Player 1 has higher skill than Player 2 using 2-d Gaussian approximation of the joint skills

```
def prob_w1_gt_w2_joint(samples1, samples2):
    mu = np.mean(samples1) - np.mean(samples2)
    cov = np.cov(samples1, samples2)
    s = np.sqrt(cov[0, 0] + cov[1, 1] - 2 * cov[0, 1])
    return norm.cdf(mu / s) if s != 0 else 0
```

Comparison of the three methods

Methods 1 and 2 assume that the marginal skill posteriors $p(w_i|\mathbf{y})$ and the joint skills posterior $p(w_i, w_j|\mathbf{y})$ are Gaussians but the true skill posteriors are not Gaussians. Method 2 is better than method 1 since it accounts for the correlation between players. Method 3 makes no assumptions on the posterior distributions; if the Gibbs sampler has converged, the skills are essentially sampled from the true joint posterior $p(\mathbf{w}, \mathbf{t}|\mathbf{y})$ and hence method 3 would produce the best results given enough samples to get reliable estimations.

Using method 3, the 4×4 table of $p(w_i > w_j|\mathbf{y})$ is computed (Table 3).

	Djokovic	Nadal	Federer	Murray
Djokovic	-	0.951	0.921	0.989
Nadal	0.049	-	0.432	0.775
Federer	0.079	0.598	-	0.811
Murray	0.011	0.225	0.190	-

Table 3: Table of probability that *one player has higher skill than the other* computed using direct samples from Gibbs sampler. Entry at row i and column j represents the probability that Player i has higher skill than Player j.

Table 3 and Table 1 (message passing) are very similar but Gibbs sampling tends to produce a greater $p(w_i > w_j|\mathbf{y})$ when $E(w_i|\mathbf{y}) > E(w_j|\mathbf{y})$. The errors in Gibbs sampling come from the calculations using limited samples. The errors in message passing come from approximating the variables $\mathbf{w}$, $\mathbf{t}$ as Gaussians. Although Gibbs sampling could give a more accurate result with large enough samples, message passing requires much less computational power and storage while producing a very good approximation.

Section e

Rankings using empirical game outcome averages (System 1)

The players are ranked based on their win-rates in the games from the dataset (Figure 12).

Rankings using Gibbs sampling (System 2)

To take into account the players' variance in skills, we rank the players based on the average winning probability assuming they played all the other players in the league once. For each pair of players (i, j), the probability that Player i beats Player j is calculated as follows:

For the nth skill sample,

$$p_{gs}(y_{i,j}^{(n)} = 1|w_i^{(n)}, w_j^{(n)}) = \Phi(w_i^{(n)} - w_j^{(n)}) \tag{4}$$

Average across all N Gibbs samples:

$$p_{gs}(y_{i,j} = 1|\mathbf{w}_i, \mathbf{w}_j) = \frac{1}{N}\sum_{n=1}^{N} \Phi(w_i^{(n)} - w_j^{(n)}) \tag{5}$$

The players are then ranked based on the average winning probability against the $(M-1)$ opponents:

$$p_{gs}(y_i = 1|\mathbf{W}) = \frac{1}{M-1} \sum_{\substack{j=1 \\ j \neq i}}^{M} p_{gs}(y_{i,j} = 1|\mathbf{w}_i, \mathbf{w}_j) \tag{6}$$

Ranking using message passing (System 3)

Using the results of message passing, we can similarly compute the winning probability of each player, averaged across all 106 opponents and rank them based on this.

$$p_{mp}(y_i = 1|\mathbf{W}) = \frac{1}{M-1} \sum_{\substack{j=1 \\ j \neq i}}^{M} \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2 + 1}}\right) \tag{7}$$

Figure 13 shows the results using Gibbs sampling and message passing.

Results and comparison

The rankings using the three different methods are compared in Figure 14.

Gibbs sampling and message passing should give virtually the same rankings since they use the same quantity (average winning probability) to rank the players. The difference is caused by approximation errors in probabilities.

The rankings produced by System 1 are much different to the rankings by TrueSkill (System 2 and System 3). System 1 only considers the binary win/loss outcomes whereas TrueSkill considers both the qualities of the oppositions and the number of matches each player has played to produce the rankings.

For example, the rankings of Nadal and Murray are (3rd, 4th) using TrueSkill and (4th, 3rd) using win-rates. Table 4 shows that although Murray has a marginally higher win-rate, Nadal played more matches and his opponents have higher skills and rankings on average. Therefore, that Nadal ranks higher than Murray in TrueSkill despite having a lower win-rate.

	Nadal	**Murray**
No. of games	72	59
Win-rate	79.2%	79.7%
Avg. opponent skill in games they won	0.43	0.27
Avg. opponent skill in games they lost	1.20	0.95
Avg. opponent ranking in games they won	28.6	35.6
Avg. opponent ranking in games they lost	9.2	18.4
Mean skill	1.48	1.28
Skill variance	0.040	0.043

Table 4: Summary of matches played by Rafael Nadal and Andy Murray. The average opponent skills and rankings are the ones calculated using Gibbs sampling outcomes.

Moreover, if multiple players have identical win-rates, System 1 would give them the same ranking while TrueSkill would consider the opponents and the number of the matches the player played to determine who ranks higher.

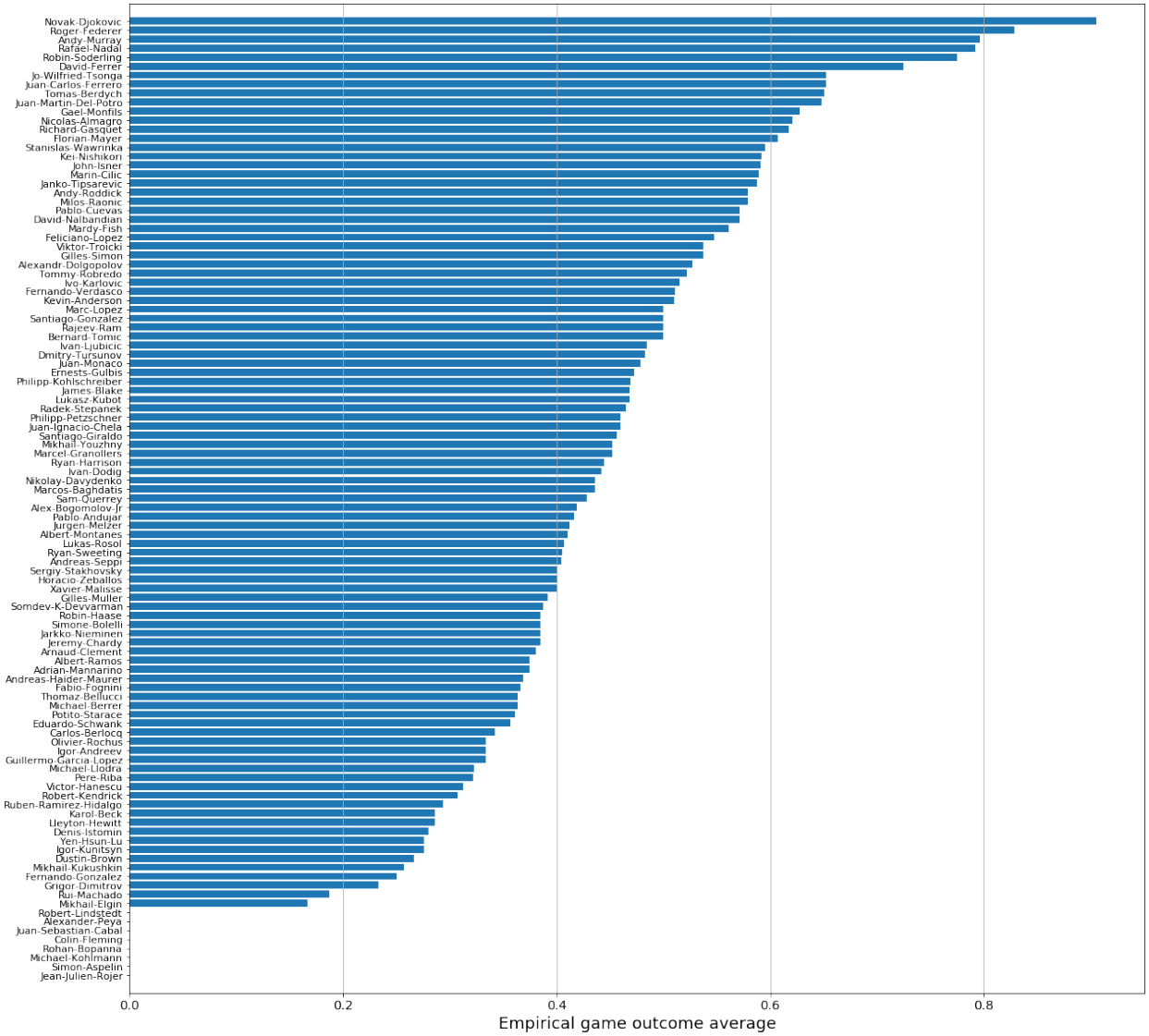


Figure 12: Ranked empirical game outcome averages. For players who have the same win-rate, this ranking system cannot tell the difference in skills between them and would give them the same ranking, regardless of their opponents and number of matches played.

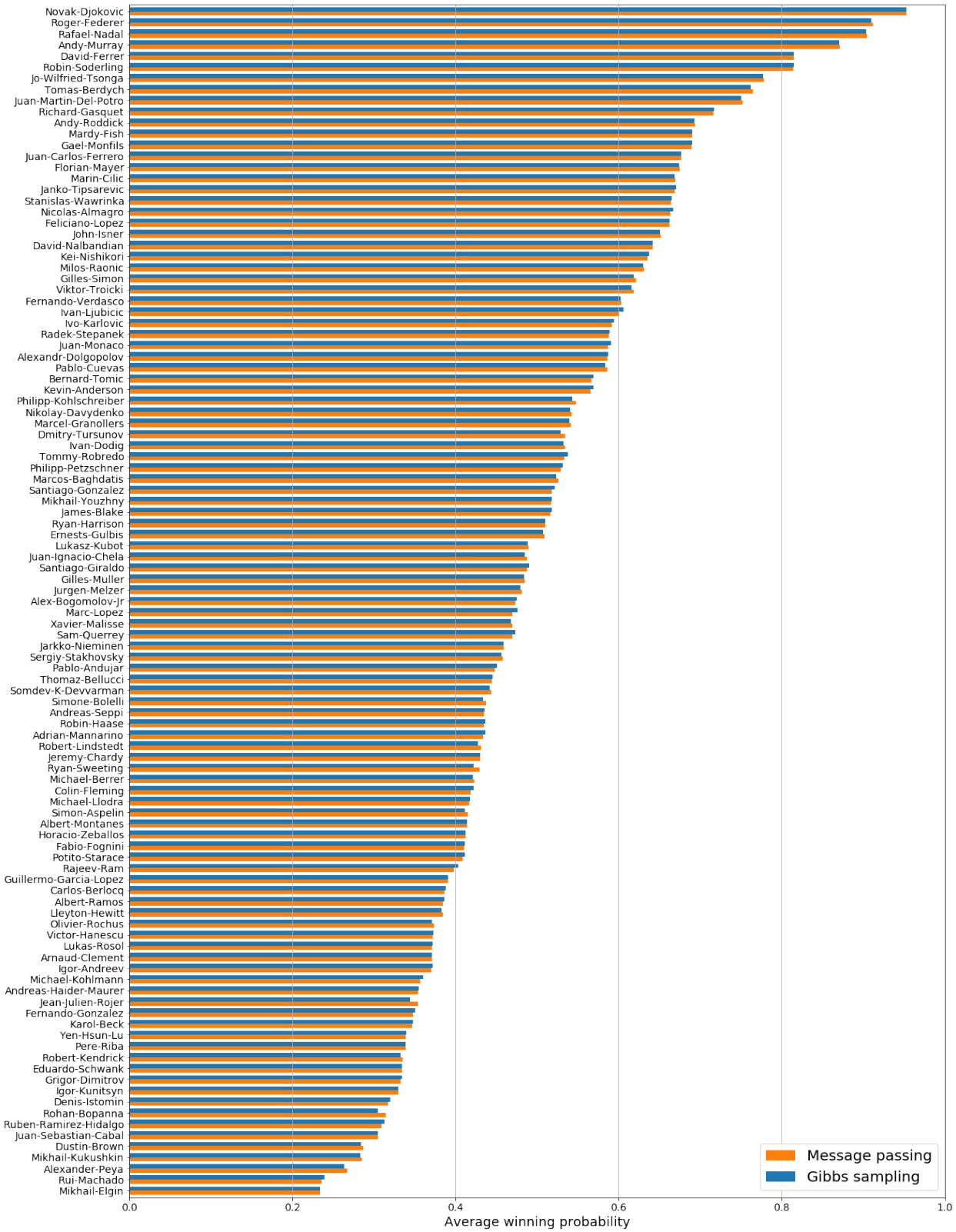


Figure 13: Ranked winning probabilities calculated using Gibbs sampling and message passing

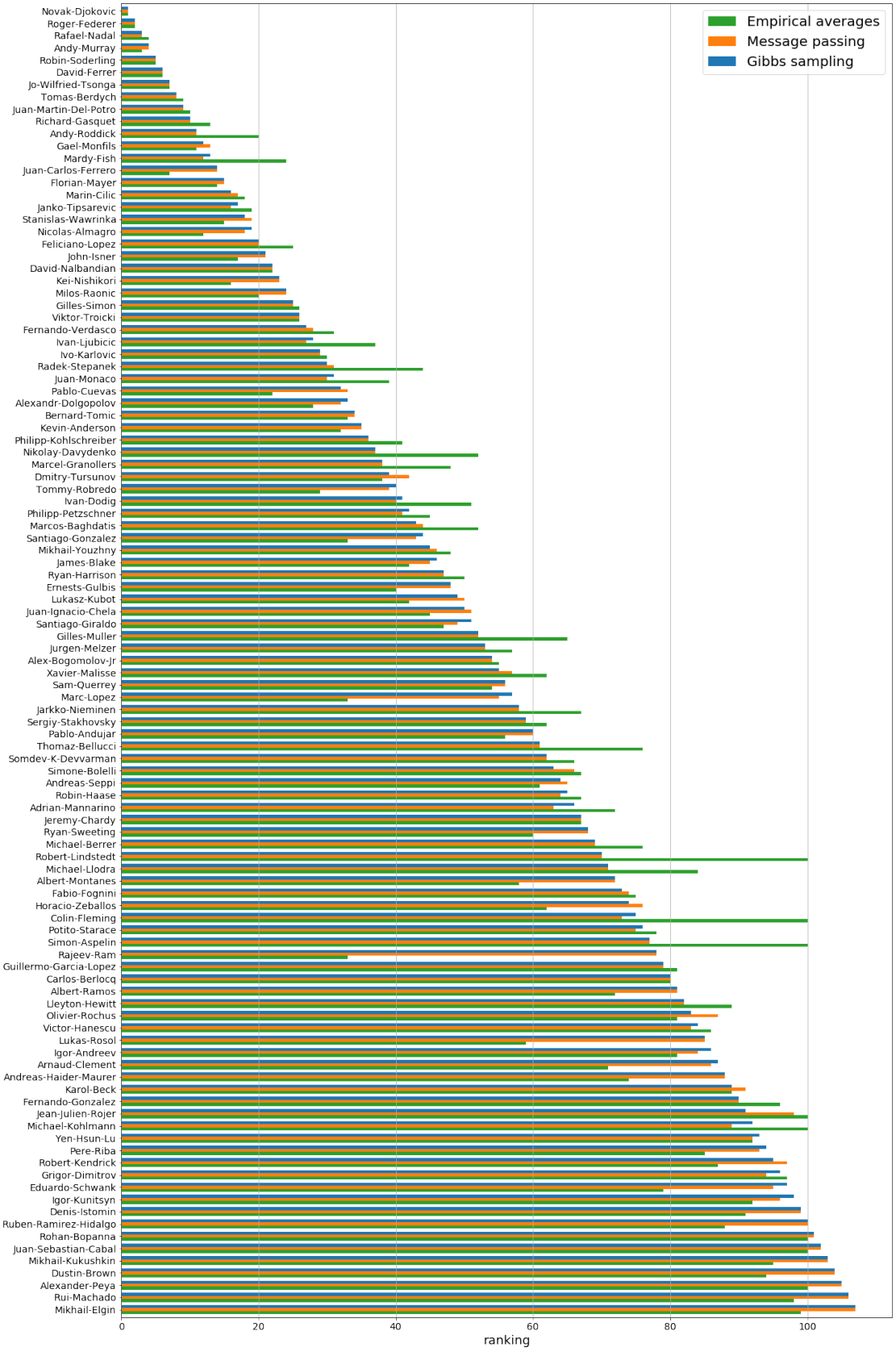


Figure 14: Rankings of the 107 players using the three different methods: average winning probability using Gibbs sampling, average winning probability using message passing, and simple win-rate

References

Geweke, J. 1992. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. *Bayesian Statistics 4*, J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (eds.), pp. 169–193. Oxford University Press.

Appendix

Derivation of probability that one player has higher skill than another, where the two players' skills are jointly distributed as a 2-d Gaussian:

$$p(w_i, w_j|\mathbf{y}) = \mathcal{N}(\mathbf{m}, \Sigma)$$
$$\mathbf{m} = \begin{bmatrix} \mu_i \\ \mu_j \end{bmatrix} \text{ and } \Sigma = \begin{bmatrix} \sigma_i^2 & \sigma_{ij} \\ \sigma_{ij} & \sigma_j^2 \end{bmatrix}$$

The skill difference $s = w_i - w_j$ is Gaussian (linear combination of Gaussians):

$$s \sim \mathcal{N}(\mu_s, \sigma_s^2)$$

Mean (conditioned on $\mathbf{y}$):

$$\mu_s = E[s] = \mu_i - \mu_j$$

Second moment (conditioned on $\mathbf{y}$):

$$\begin{aligned} E[s^2] &= E[(w_i - w_j)^2] \\ E[s^2] &= E[w_i^2 + w_j^2] - 2E[w_i w_j] \\ E[s^2] &= E[w_i^2 + w_j^2] - 2\sigma_{ij} - 2\mu_i\mu_j \end{aligned}$$

where σ_{ij} is the covariance between sequence i and sequence j.

Variance:

$$\begin{aligned} \sigma_s^2 &= E[s^2] - E[s]^2 = E[s^2] - (\mu_i - \mu_j)^2 \\ \sigma_s^2 &= E[w_i^2 + w_j^2] - 2\sigma_{ij} - 2\mu_i\mu_j - \mu_i^2 + 2\mu_i\mu_j - \mu_j^2 \\ \sigma_s^2 &= \sigma_i^2 + \sigma_j^2 - 2\sigma_{ij} \end{aligned}$$

$$\therefore \; \underline{\underline{s \sim \mathcal{N}(\mu_i - \mu_j, \sigma_i^2 + \sigma_j^2 - 2\sigma_{ij})}}$$