Probabilistic Ranking

Probabilistic Machine Learning

Abhishek Shenoy

Total words = 999
[Excluding tables, figures, code, equations, abstract and references]

Abstract

This report explores probabilistic methods for ranking tennis players specifically focusing on the use of Gibbs sampling (GS) and message passing (MP) with expectation propagation (EP).

1 Part A

GS is used to draw sample skills w_p for player p. Consider $t = w_i - w_j + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$ and all the game outcomes as a binary vector $\boldsymbol{y}$. For each iteration i, we draw a sample $\boldsymbol{t}^{(i)}$ from the conditional distribution $p(\boldsymbol{t}|\boldsymbol{w}^{(i-1)}, y)$ and then draw a sample of $\boldsymbol{w}$ from the Gaussian conditional distribution $p(\boldsymbol{w}|\boldsymbol{t}^{(i)})$.

```
## Update mean and covariance
for g in range(N):
    winner, loser = G[g]

    m[winner] += t[g]
    m[loser] -= t[g]

    iS[winner, loser] -= 1
    iS[loser, winner] -= 1

    iS[winner, winner] += 1
    iS[loser, loser] += 1
```

Listing 1. Update of mean and covariance for conditional Gaussian distribution $p(\boldsymbol{w}|\boldsymbol{t}^{(i)})$ for GS

1.1 Burn-in

When using GS, the initial samples are dependent on the initialisation and are not drawn from the target posterior hence we discard a number of initial samples known as burn-in.

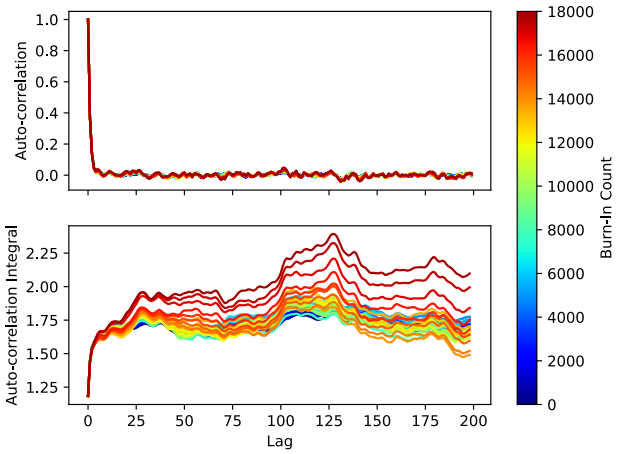

Figure 1. Auto-correlation and its integral from the skill samples using GS for different burn-in values for player-1

From Figure 1 we notice that the auto-correlation integral stays most steady-state at a burn-in value around 4000. By generating samples for a range of players (Fig 3), we can verify that the distributions are independent.

1.2 Thinning

To mimic independent samples drawn from the posterior we can keep every n-th sample, where samples $w^{(i)}$ and $w^{(i+n)}$ have no correlation. When testing the convergence of MCMC methods, we can consider the first lag value at which the auto-correlation equals 0 [1]. Figure 2 shows stationarity for the auto-correlation integral and zero-intercept for auto-correlation at around 10 suggesting a thinning value of 10 however we simply use 20 to further ensure reduced dependence. Alternatively we could perform convergence tests such as the Gelman-Rubin test [2, 3] but here it is clear that the steady-state is reached at a thinning value of 10 (and more convincingly at 20 as in Fig 1).

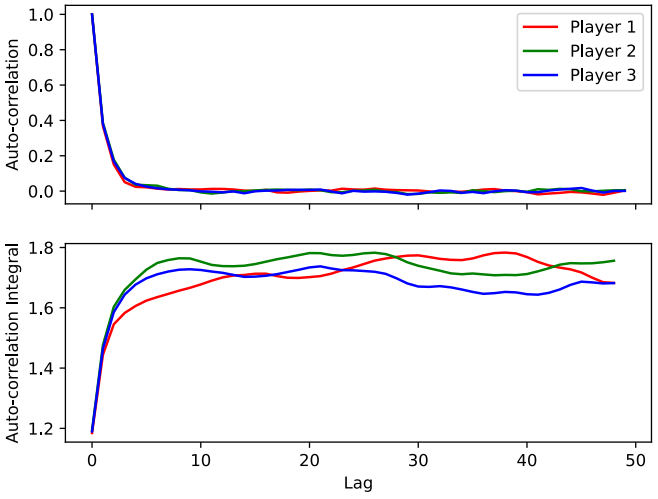

Figure 2. Auto-correlation and its integral from the skill samples using GS for different players

1.3 Testing

We generated 24,000 samples and found that 4000 samples was a suitable burn-in and used a thinning value of 20 giving 1000 independent samples. Figure 3 shows that the samples drawn for the 3 different players are from independent distributions (with different means and variances).

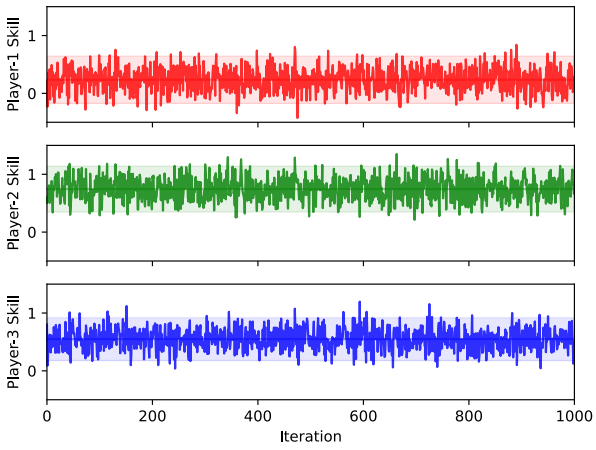

Figure 3. Skill samples drawn using GS with μ line and $\pm 2\sigma$ range with burn-in and thinning

2 Part B

2.1 Gibbs Sampling

GS is used to obtain independent samples from the intractable joint distribution. When the sample distribution converges (has steady-state mean and variance), we can interpret the sample as being drawn from the joint distribution. Convergence occurs when the samples appear to be drawn from a fixed posterior distribution. Finding the exact point of convergence is difficult but by plotting kernel density estimates using an incremental number of samples (starting with 100 with increments of 100) (Fig 4), we notice the distribution starts to converge after around 4000 samples. This is also evidence for the burn-in to discard the first 4000 samples for our dataset of 24000 samples. Using more samples has no significant impact on the distribution after around 12000 samples.

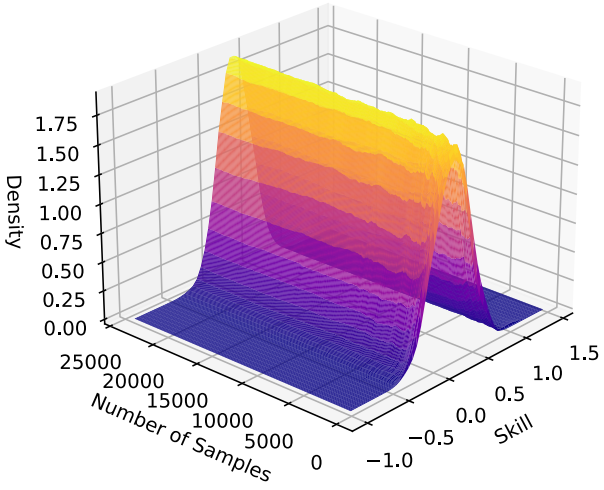

Figure 4. Kernel density estimates for various number of samples without burn-in or thinning for player-1 showing convergence

2.2 Message Passing

For MP the marginals are assumed to be Gaussian. Therefore, only the means and variances are used as the message. To measure convergence for MP, we can consider the means and variances and find the iteration where the distribution mean changes by less than a threshold or becomes steady-state. Figure 5 shows that the point of convergence is approximately 50 iterations.

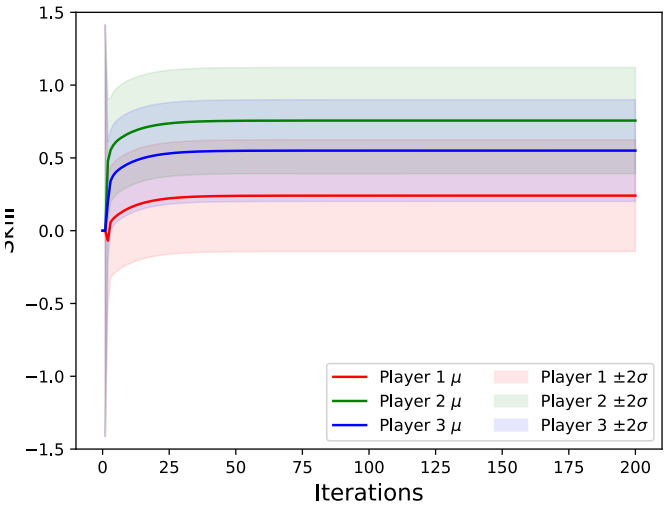

Figure 5. Mean and standard deviation over iterations of MP

3 Part C

The top four players by the ATP ranking of 2011 were Djokovic (15), Nadal (0), Federer (4) and Murray (10) in order. We evaluate these rankings by assuming Gaussian distributions characterised by the means and variances of the skill samples generated by MP. For 2 players A and B, the skill distributions are $w_A \sim \mathcal{N}(0,1)$ and $w_B \sim \mathcal{N}(0,1)$ respectively assuming independence.

```
ids = [15, 0, 4, 10]
tab1, tab2 = np.zeros((4,4)), np.zeros((4,4))

for i in range(4):
    for j in range(4):
        mean_i = mean_player_skills[ids[i]]
        mean_j = mean_player_skills[ids[j]]

        pre_i = precision_player_skills[ids[i]]
        pre_j = precision_player_skills[ids[j]]

        mean = mean_i - mean_j
        variance1 = 1/pre_i + 1/pre_j
        variance2 = 1/pre_i + 1/pre_j + 1

        scale1 = variance1**0.5
        scale2 = variance2**0.5

        prob1_j = scipy.stats.norm.cdf(0, mean, scale1)
        prob2_j = scipy.stats.norm.cdf(0, mean, scale2)

        tab1[j, i] = prob1_j
        tab2[j, i] = prob2_j

print(tab1)
print(tab2)
```

Listing 2. Probability tables for top players using MP

3.1 Greater skill

The probability of player A having a greater skill than B can be calculated using Equation 1.

$$
\begin{aligned}
P(w_A > w_B) &= P(w_A - w_B > 0) \\
&= \int_0^\infty \mathcal{N}(\mu_A - \mu_B, \sigma_A^2 + \sigma_B^2) \\
&= \phi\Big(- \frac{\mu_A - \mu_B}{\sqrt{\sigma_A^2 + \sigma_B^2}} \Big)
\end{aligned} \tag{1}
$$

A \ B	15	0	4	10
15	0.5	0.940	0.909	0.985
0	0.060	0.5	0.427	0.767
4	0.091	0.573	0.5	0.811
10	0.015	0.233	0.189	0.5

Table 1. Probability of player A having a higher skill than player B using the MP distribution

3.2 Match win

We can define the performance uncertainty as a standard Gaussian $\epsilon \sim \mathcal{N}(0, 1)$. The probability of player A winning against B can then be calculated using Equation 2.

$$
\begin{aligned}
P(w_A - w_B > \epsilon) &= P(w_A - w_B - \epsilon > 0) \\
&= \int_0^\infty \mathcal{N}(\mu_A - \mu_B, \sigma_A^2 + \sigma_B^2 + 1) \\
&= \phi\Big(- \frac{\mu_A - \mu_B}{\sqrt{\sigma_A^2 + \sigma_B^2 + 1}} \Big)
\end{aligned} \tag{2}
$$

Table 2 shows the probabilities of the match outcomes favouring the weaker player and bringing the probabilities closer to 0.5. This is due to the added uncertainty from ϵ being added to the greater-skill equation (Eq 1) giving Equation 2. The large variance of 1 in the noise variable relative to the marginal skill uncertainties favours the weaker player.

A \ B	15	0	4	10
15	0.5	0.655	0.638	0.720
0	0.345	0.5	0.482	0.573
4	0.362	0.518	0.5	0.591
10	0.280	0.427	0.409	0.5

Table 2. Probability of player A winning against player B using the MP distribution

4 Part D

Here we compare the skills (generated by GS) of Djokovic (w_1) and Nadal (w_2) using three different methods. 4000 of 24000 samples are discarded (burn-in) and 1 in 20 samples are kept (thinning) giving a total of 1000 reliable samples.

4.1 Approximate marginal skills by Gaussians ($\mathcal{M}1$)

For each player, the mean and variance are calculated empirically using Equation 3. $P(w_1 > w_2)$ is then calculated using Equation 1 assuming that each skill has a Gaussian distribution.

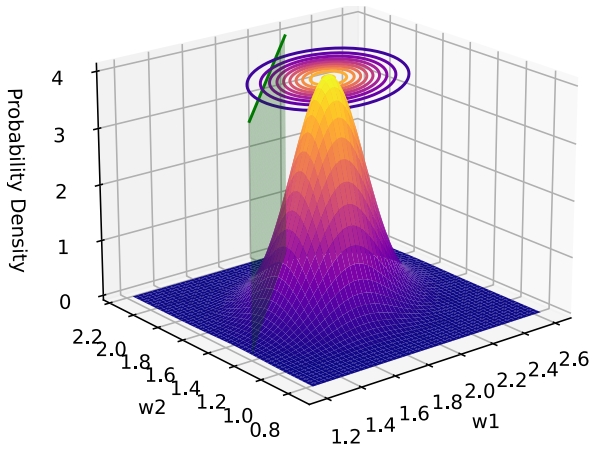

Figure 6. Bivariate joint Gaussian distribution from Djokovic and Nadal skill samples with plane denoting w1=w2

$$
\begin{aligned}
\mu_p &= \frac{1}{N} \sum_{i=0}^{N-1} w_p^{(i)} \\
\sigma_p^2 &= \frac{1}{N} \sum_{i=0}^{N-1} (w_p^{(i)} - \mu_p)^2
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
\mu_1 &= 1.88 & \sigma_1^2 &= 0.046 \\
\mu_2 &= 1.48 & \sigma_2^2 &= 0.036
\end{aligned}
$$

$$
\mathbf{P(w_1 > w_2) = 0.921}
$$

4.2 Approximate joint skills by Gaussian ($\mathcal{M}2$)

The mean and variance of the joint distribution can be calculated using Equation 4. Note that the mean calculation is the same as for $\mathcal{M}1$.

$$
\begin{aligned}
\mu_p &= \frac{1}{N} \sum_{i=0}^{N-1} w_p^{(i)} \\
\sigma_{pq}^2 &= \frac{1}{N} \sum_{i=0}^{N-1} (w_p^{(i)} - \mu_p)(w_q^{(i)} - \mu_q)
\end{aligned} \tag{4}
$$

$$
\mu = \begin{bmatrix} 1.88 \\ 1.48 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 0.046 & 0.011 \\ 0.011 & 0.036 \end{bmatrix} \qquad \mathbf{P(w_1 > w_2) = 0.950}
$$

The volume under the bivariate Gaussian (Fig 6) on the side of the plane for which $w_1 > w_2$ denotes the probability of Djokovic having a greater skill than Nadal.

4.3 Direct samples ($\mathcal{M}3$)

$$
P(w_1 > w_2) \approx \frac{1}{N} \sum_{i=0}^{N-1} \mathbb{1}_{[w_1 > w_2]}(w_1^{(i)}, w_2^{(i)}) \tag{5}
$$

$$
\mathbf{P(w_1 > w_2) = 0.955}
$$

```python
## Direct Samples
ps = [15, 0, 4, 10]
w1 = skill_samples[ps[0], burn_in::thin]
w2 = skill_samples[ps[1], burn_in::thin]

# Prob of Djokovic > Nadal skill
prob = sum(w1 > w2)/len(w1)
print(prob)
```

Listing 3. Probability of Djokovic having greater skill than Nadal using direct samples from GS

4.4 Best Method

Both $\mathcal{M}1$ and $\mathcal{M}2$ impose a Gaussian distribution. Furthermore $M1$ does not account for dependence between the skills. Therefore $\mathcal{M}3$ is most suitable as it directly uses the samples generated.

A \ B	15	0	4	10
15	0	0.955	0.908	0.993
0	0.045	0	0.443	0.807
4	0.092	0.557	0	0.815
10	0.007	0.193	0.185	0

Table 3. Probability of player A having a higher skill than player B directly using the Gibbs samples ($\mathcal{M}3$)

Comparing the probabilities in Table 3 (GS) to Table 1 (MP), we notice that the GS probabilities are very close to the EP approximation using MP.

```python
## Direct Samples
ps = [15, 0, 4, 10]
w1 = skill_samples[ps[0], burn_in::thin]
w2 = skill_samples[ps[1], burn_in::thin]
w3 = skill_samples[ps[2], burn_in::thin]
w4 = skill_samples[ps[3], burn_in::thin]

ws = [w1, w2, w3, w4]
num_players = len(ps)
num_samples = len(ws[0])
prob_matrix = np.zeros((num_players, num_players))

for i in range(num_players):
    for j in range(num_players):
        prob_matrix[i, j] = sum(ws[i] > ws[j])/
    num_samples
print(prob_matrix)
```

Listing 4. Probability of greater skill for top players using direct samples from GS

5 Part E

Here we compare the player ranks from 3 different inference methods. The empirical method ranks players based upon the fraction of games won. GS and MP give almost identical ranks (differing by maximum 2) verifying the performance of the EP approximation (Fig 7).

However GS requires a large number of samples for reliable estimates of the mean and variance and their values fluctuate depending on the new samples. In contrast, MP aims to optimise the mean and variance directly hence converging quicker and independent of initial conditions. Therefore MP must be preferred to GS.

In Figure 8, when we plot the ranks as separate dimensions, we see that the differences are majoritively due to the variation in empirical rank. The green line is the line for which all 3 methods would agree with each other.

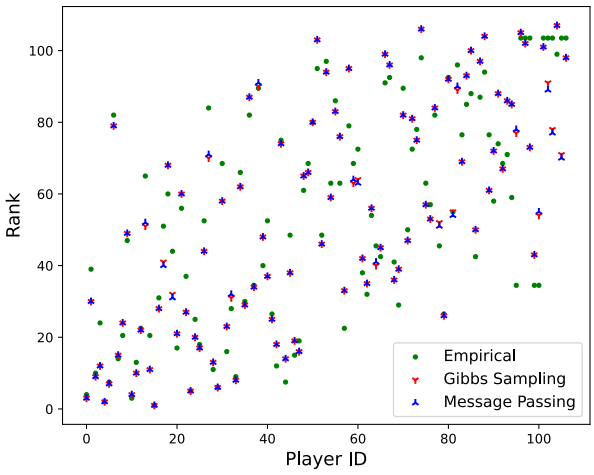

Figure 7. Player ranks using different methods

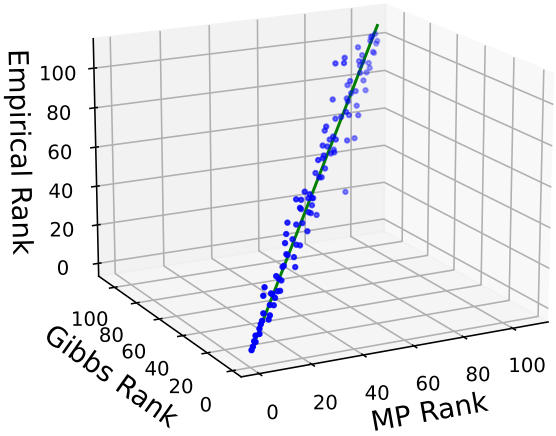

Figure 8. Rank variation between methods

```python
## Empirical Rank
# count wins and total games for each player
game_counts = np.zeros((M,2))
for g in range(N):
    w, l = G[g]
    game_counts[w][0] += 1
    game_counts[w][1] += 1
    game_counts[l][1] += 1
win_props = [game_counts[p][0]/game_counts[p][1] if
    game_counts[p][1] != 0 else 0 for p in range(M)]
empranks = len(win_props) - rankdata(win_props, method='
    average') + 1


## Gibbs Rank
skill_samples = np.load('skill_samples_24k.npy')
burn_in, thin = 4000, 20
gibbs_means = [skill_samples[p, :][burn_in::thin].mean()
    for p in range(len(skill_samples))]
gibranks = len(gibbs_means) - rankdata(gibbs_means,
    method='average') + 1


## MP/EP Rank
num_iters = 200
# run message passing algorithm, returns mean and
    precision for each player
mean_player_skills, precision_player_skills, allMs, allPs
    = eprank(G, M, num_iters)
epranks = len(mean_player_skills) - rankdata(
    mean_player_skills, method='average') + 1


## Plot all 3 ranks for each player index
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
scat = ax.scatter3D(epranks, gibranks, empranks)
plt.show()
```

Listing 5. Rank comparison code

By now considering only the 2 dimensions MP and empirical rank (Fig 9), we clearly see the inaccurate rank estimates of the empirical method. At the best ranks (lowest rank values) there is agreement, whilst there is significant inconsistency at the worst ranks (highest rank values).

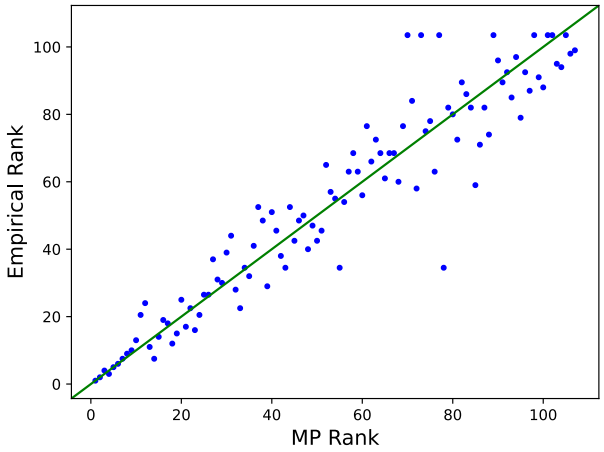

Figure 9. Empirical vs MP rank comparison

The best players tend to have played the most games reducing the significance of the opponent's skill distribution. Hence the empirical method can succeed without considering the opponent. However at the worst ranks there are fewer games played, so more information could have been gained by considering the opponent. Hence, the empirical method suffers and deviates from MP and GS rankings as it will overestimate a weak player since the strength of the opponent is disregarded.

In contrast, probabilistic approaches consider the opposing player's skill, permitting more meaningful ranks for each player.

The limitation of MP are the assumptions that the factor graph is a tree and moment matching is reasonable in approximating the performances distribution however it converges much faster than GS.

References

[1] E. Ford. (2015) MCMC Diagnostics. [Online]. Available: https://astrostatistics.psu.edu/RLectures/diagnosticsMCMC.pdf

[2] E. Sellentin. (2018) Convergence tests for MCMC. [Online]. Available: https://www.imperial.ac.uk/media/imperial-college/research-centres-and-groups/astrophysics/public/icic/data-analysis-workshop/2018/Convergence-Tests.pdf

[3] S. Sinharay. (2003) MCMC Convergence. [Online]. Available: https://www.ets.org/Media/Research/pdf/RR-03-07-Sinharay.pdf