Gwangbin Bae gb585@cam.ac.uk

4F13: Probabilistic Machine Learning
Coursework #2: Probabilistic Ranking

1. Question A

Command 1 *Mean of the conditional skills*

```
m(p) = t'*((p==G(:,1)) - (p==G(:,2)));
```

Command 2 *Sum of precision matrices*

```
if i==j
    iS(i,j) = sum(i==G(:,1)) + sum(i==G(:,2));
else
    iS(i,j) = -sum((i==G(:,1)).*(j==G(:,2))+(i==G(:,2)).*(j==G(:,1)));
    iS(j,i) = iS(i,j);
```

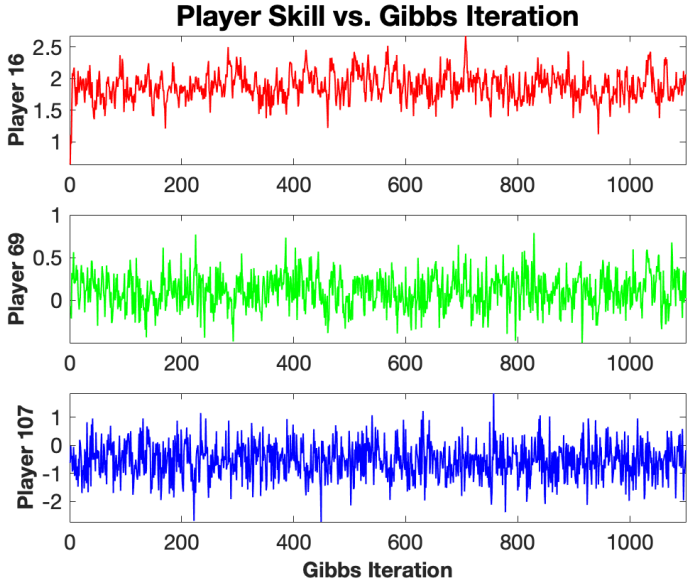

Figure 1 *Player skill, plotted against the Gibbs iteration*

Figure 1 shows the skills of three different players at each Gibbs iteration. For short time period ($\Delta t < 50$), the behaviour of the plots (mean and variance) changes significantly. However, in macroscopic perspective ($\Delta t > 100$), the distribution seems stationary (i.e. having fixed mean and variance). This suggests that the Gibbs sampler is moving around the whole posterior.

The mixing time of the Gibbs sampler is the amount of time required to reach the steady state distribution. Depending on the initial condition (e.g. random seeds), the starting point of the Gibbs sampler can be far from the desired distribution, resulting in long mixing time. Hence, it is often advised to discard the initial samples. The number of discarded initial samples is often called the 'burn-in period'.

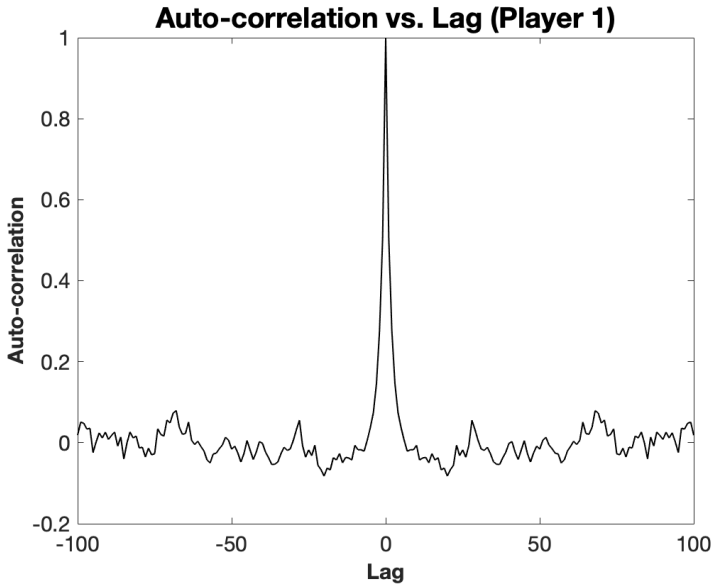

Figure 2 *Auto-correlation, plotted against the lag*

Figure 2 shows the auto-correlation plotted against the time lag. The area under this plot defines the effective correlation length. (The negative area must be converted to its absolute value.) The average and standard deviation of the effective correlation length are as following.

$$\mu_l = 7.4706 \ , \ \sigma_l = 0.8657$$

Hence, it is reasonable to take only every 10th sample. Such process is called 'thinning'. **Figure 3** shows the 100 samples obtained by applying 'burn-in' and 'thinning' to the original 1100 samples.

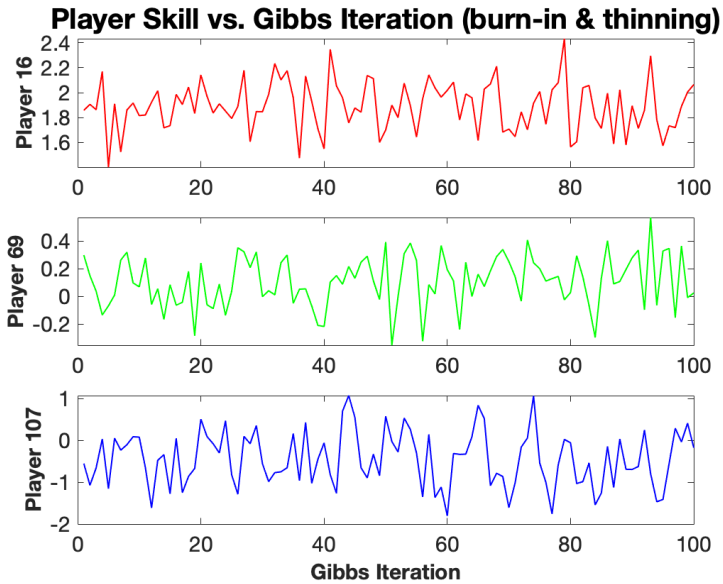

Figure 3 *Player skill, plotted against the Gibbs iteration (burn-in & thinning applied)*

2. Question B

In Gibbs sampler, the goal is to obtain independent samples that can represent the intractable joint distribution. Under such framework, convergence is achieved when the behaviour of the sample distribution (e.g. mean and variance) stays steady for a long period of time. Then, the converged sample distribution can be interpreted as the sample drawn from the joint distribution.

Since the Gibbs sampler is converging the distribution itself, it is difficult to find the precise moment at which the convergence is achieved. An alternative method can be to compare the chunks of distribution and find the chunk from which the distribution statistics (e.g. mean and variance) stay consistent.

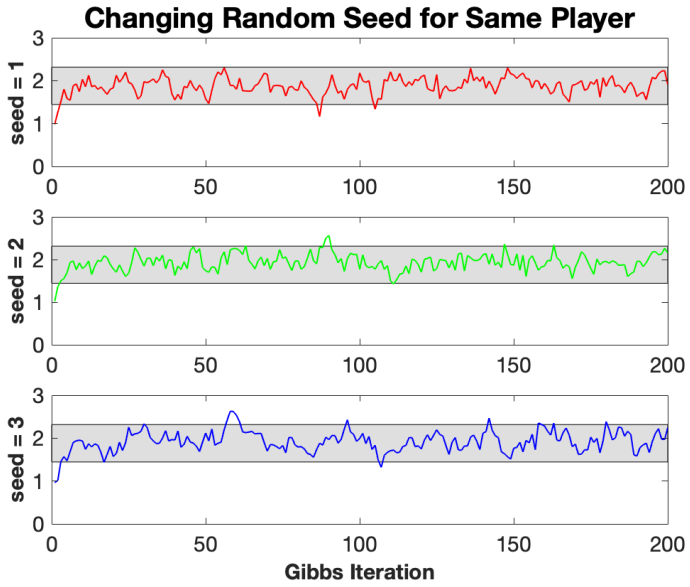

Figure 4 *The shaded area represents $\mu \pm 2\sigma$ of the converged distribution*

As can be seen in **Figure 4**, changing the initial condition does not affect the converged distribution. This is because the Gibbs sampler, after mixing time, moves around the same posterior. (Exception can occur when the joint distribution is of complex shape such as mixture of Gaussians.)

Command 3 *Message passing and EP - recording the mean and covariance at each iteration*

```
Mean_mat(:,iter) = Ms;     % mean matrix at each iteration
Cov_mat(:,iter) = 1./Ps;   % covariance matrix at each iteration
```

In message passing, the marginals are assumed to be Gaussian. Therefore, only the means and variances (and their variation such as precisions and natural means) are used as the message.

For example, the mean and variance of the marginal skills, $q^{\tau}(w_i)$ are used to update the marginal performances, $q^{\tau+1}(t_g)$, which is again used to update the marginal skills $q^{\tau+1}(w_i)$. Under such framework, convergence is achieved when:

$$q^{\tau}(w_i) = q^{\tau+1}(w_i) \quad \text{and} \quad q^{\tau}(t_g) = q^{\tau+1}(t_g)$$

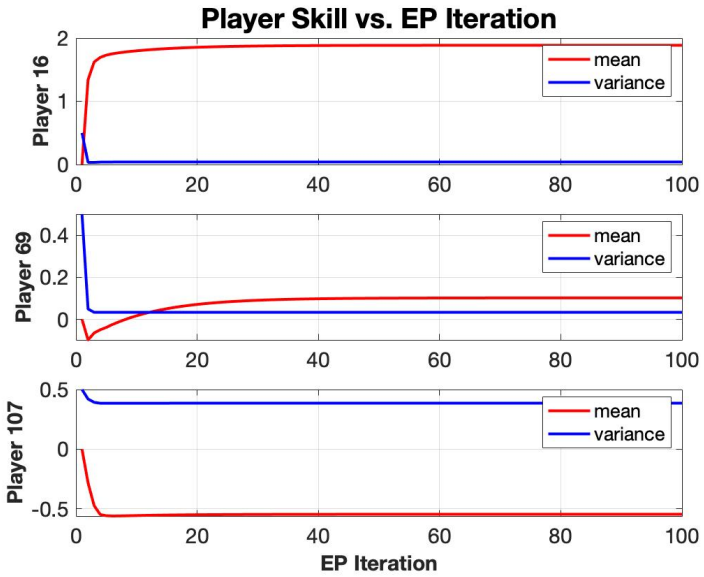

Figure 5 *Player skill, plotted against the EP iteration*

Figure 5 shows that the mean and variance of the player skills converge quickly. The number of iterations required for convergence can be obtained by defining the condition for convergence. Following can be an example.

$$q^{\tau+1}(w_i) - q^{\tau}(w_i) < 10^{-3} \quad \text{for all } i \quad \text{AND} \quad q^{\tau+1}(t_g) - q^{\tau}(t_g) < 10^{-3} \quad \text{for all } g$$

Lastly, in order to visualise the effect of initialisation on convergence, the message passing algorithms are repeated for different initial values.

Command 4 *Changing the initial condition for message passing*

```
Mgs = zeros(N,2) + k; % k = 0, 0.5, 1.0, 2.0
Pgs = zeros(N,2) + k;
```

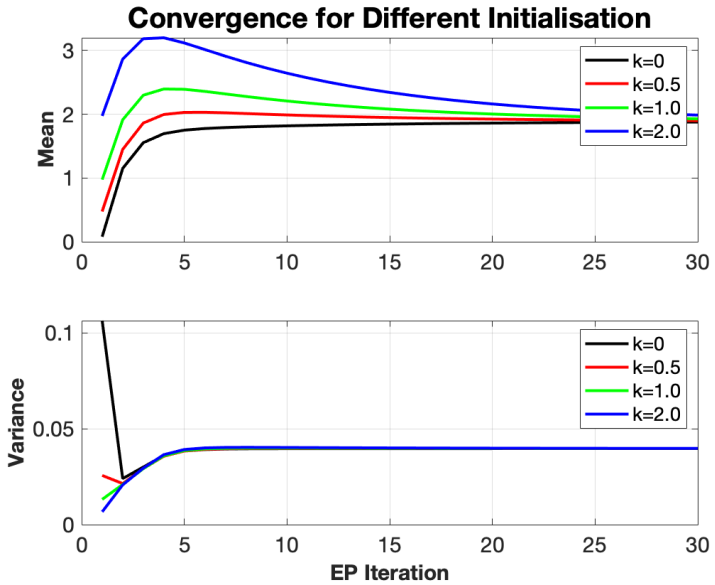

Figure 6 *Convergence for different initialisation*

In **Command 4**, the matrices `Mgs` and `Pgs` represent the game to skill message. Instead of initialising these messages with zeros, a small number k is added. **Figure 6** shows that the convergence is independent of the initial condition. This is mainly because the message passing algorithms eventually tries to fit itself to the data.

3. Question C

Command 5 *Calculating the probability that the skill of player 1 is higher than that of player 2*

```
skill_prob(i,j) = 1 - normcdf(0, Mu(i)-Mu(j), sqrt(Var(i)+Var(j)))
```

Table 1 *Probabilities that the skill of Player 1 is higher than that of Player 2 (Message Passing)*

		Player 2			
		Novak Djokovic	*Rafael Nadal*	*Roger Federer*	*Andy Murray*
Player 1	*Novak Djokovic*		0.9398	0.9089	0.9853
	Rafael Nadal	0.0602		0.4271	0.7665
	Roger Federer	0.0911	0.5729		0.8108
	Andy Murray	0.0147	0.2335	0.1892	

The reasoning behind Command 5 is as follows: $p(w_1 > w_2)$ is equivalent to $p(w_1 - w_2 > 0)$. By setting $z = w_1 - w_2$, $p(z > 0)$ can be obtained as following.

$$z = w_1 - w_2 = \mathcal{N}(z; \mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2)$$

$$p(z > 0) = \int_0^\infty \mathcal{N}(z; \mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2) = 1 - \int_{-\infty}^0 \mathcal{N}(z; \mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2)$$

Command 6 *Calculating the probability of Player 1 winning against Player 2*

```
winning_prob(i,j)= normcdf((Mu(i)-Mu(j))/sqrt(1 + Var(i) + Var(j)));
```

Table 2 *Probabilities of Player 1 winning against Player 2 (Message Passing)*

		Player 2			
		Novak Djokovic	*Rafael Nadal*	*Roger Federer*	*Andy Murray*
Player 1	*Novak Djokovic*		0.6554	0.6380	0.7198
	Rafael Nadal	0.3446		0.4816	0.5731
	Roger Federer	0.3620	0.5184		0.5909
	Andy Murray	0.2802	0.4059	0.4245	

Table 1 and **2** represent highly correlated probabilities. For example, if Player 1 is likely to have more skill than Player 2, he is likely to win against him as well. Therefore, the 'yes's and 'no's match in the two tables. However, the probabilities in Table 2 are much closer to 0.5. In other words, the confidence in guessing who will win is much smaller compared to the confidence in guessing who is more skilled player.

This is because of the noise term added to the skill difference. Noise makes the game 'fairer'. For example, if the noise variance is much bigger than skills, the winner would be decided almost purely by the noise (coin-toss situation). On the other hand, if the noise is zero, the two tables would have same entries:

$$p(y = 1) = p(t > 0) = p(w_1 - w_2 > 0) = p(w_1 > w_2)$$

4. Question D

***NOTE**: Comparison between Player 1 (Nadal), and Player 16 (Djokovic) is incorporated in the below discussion since they are part of the 4-by-4 tables.*

***Table 3** Probabilities that the skill of Player 1 is higher than that of Player 2 (Gibbs Sampling)*

		Player 2			
		Novak Djokovic	*Rafael Nadal*	*Roger Federer*	*Andy Murray*
Player 1	*Novak Djokovic*		0.9536	0.9131	0.9706
	Rafael Nadal	0.0464		0.4081	0.7115
	Roger Federer	0.0869	0.5919		0.7632
	Andy Murray	0.0294	0.2885	0.2368	

***Table 4** Probabilities that the skill of Player 1 is higher than that of Player 2 (Gibbs Sampling, Direct)*

		Player 2			
		Novak Djokovic	*Rafael Nadal*	*Roger Federer*	*Andy Murray*
Player 1	*Novak Djokovic*		0.9703	0.8713	0.9901
	Rafael Nadal	0.0297		0.3663	0.7822
	Roger Federer	0.1287	0.6337		0.8812
	Andy Murray	0.0099	0.2178	0.1188	

Table 3 is obtained by applying **Command 5** to the mean and variance of the sampled skills. The entries show strong agreement with **Table 1** generated using message passing. This suggests that the mean and variance optimised using message passing correspond well with the samples generated using Gibbs sampling.

Table 4 is obtained directly from the joint samples. The entries are therefore essentially the maximum-likelihood estimates. It counts the number of corresponding events and sets the probability accordingly. Hence, the values are highly sample-dependent.

Also, the numbers in **Table 4** are obtained simply by binary counting (which is bigger). Thus, the model cannot incorporate the skills of each player. **Table 3**, on the other hand, computes the mean and variance of each player's skill. Then, by assuming the skills are Gaussian, the model calculates the required probability. Such method is tractable and interpretable, and thus should be preferred.

5. Question E

***Command** 7 Horizontal bar plot for ranking*

```
[kk, ii] = sort(mean_prob, 'descend');
barh(kk_g(np:-1:1))
set(gca, 'YTickLabel', W(ii_g(np:-1:1)), 'YTick', 1:np, 'FontSize', 5)
```

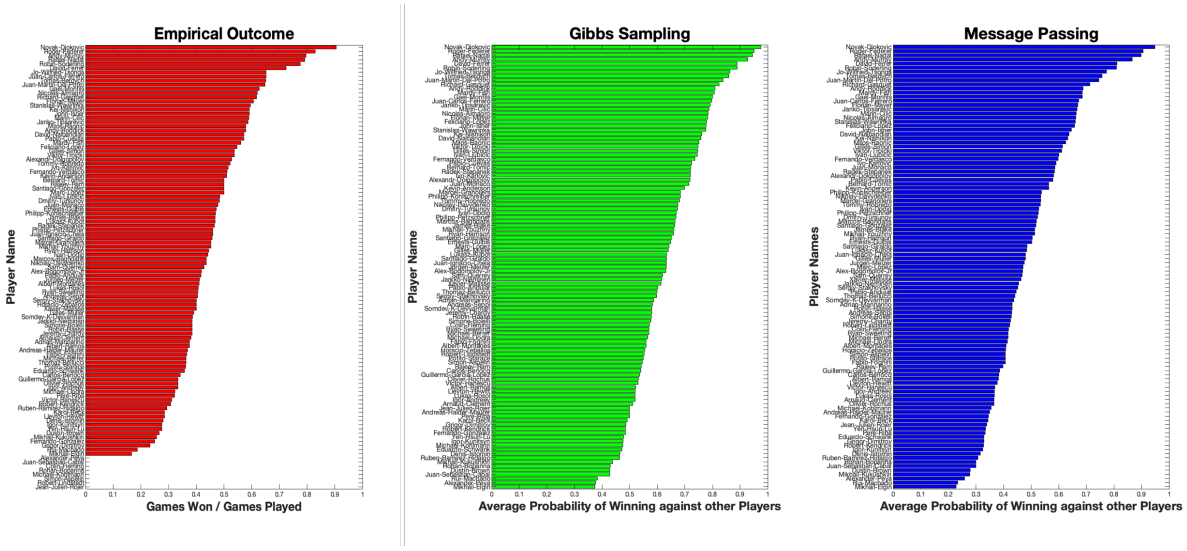

***Figure** 7 The rankings based on three algorithms – empirical outcome, Gibbs sampling, message passing*

Figure 7 shows the three rankings obtained using three different algorithms. The first algorithm simply counts the games won and divide it with the games played. Such method is not appropriate since it does not account for the level of the opponent. Also, under such framework, ranking is highly unstable (a new player who wins the first game will have the highest ranking).

Second plot shows the ranking generated with Gibbs sampling. In order to incorporate probabilistic interpretation, the average probability of winning against other players is computed using **Command 6**. The third plot is generated similarly but with message passing.

***Table 5** Ranking of the players who has not won any game, based on message passing*

Player	**R. Lindstedt**	**C. Fleming**	**S. Aspelin**	**M. Kohlmann**	**J. J. Rojer**	**R. Bopanna**	**J. S. Cabal**	**A. Peva**
Ranking	68	69	75	88	92	101	102	105

It can be seen from **Table 5** that the players who has not won any game has gained some ranking based on the consideration on who they 'lost' against.

In the second and third plot, the ranking of each player is nearly the same. (The difference is generally less than 3.) As explained earlier, this suggests that the mean and variance optimised using the message passing correspond well with the sample mean and variance obtained from Gibbs sampling.

The main difference between the two algorithms is the time required for convergence. In Gibbs sampling, the mean and variance of the marginal skills are indirectly calculated from the samples. In order to obtain reliable estimates, a large number of samples must be taken. Despite such effort, the mean and variance are not fixed, but fluctuate depending on the incoming samples.

On the contrary, message passing aims to optimize the mean and variance directly. Hence, the convergence is much quicker, and the optimized values are robust under changing initial conditions. For such reason, the ranking generated from the message passing must be preferred.