1 Modeling

$$p(s_1) = N(s_1, \mu_{s_1}, \Sigma_{s_1}) \tag{1}$$

$$p(s_2) = N(s_2, \mu_{s_2}, \Sigma_{s_2}) \tag{2}$$

$$p(t|s) = N(t, s_1 - s_2, \Sigma_t) \tag{3}$$

$$p(y|t) = \text{sign}(t) \tag{4}$$

so we have:

$$p(s_1, s_2, t, y) = p(s_1)p(s_2)p(t|s_1, s_2)p(y|t) \tag{5}$$

2 Bayesian Network

Two conditionally independent sets of variables:

$$s_1 \perp y|t, \;\; s_2 \perp y|t \tag{6}$$

and the Bayesian Network is:

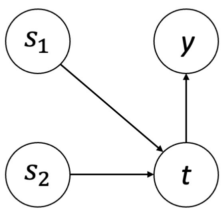

Figure 1: Bayesian Network

Denote $s = \{s_1, s_2\}$

$$p(s_1, s_2, y|t) = \frac{p(s_1, s_2, t, y)}{p(t)} = \frac{p(y|t)p(s|t)p(s)}{p(t)} = p(y|t)p(s|t) \tag{7}$$

3 Computing with the model

3.1 $p(s_1, s_2|t, y)$

let $s = \{s_1, s_2\}$

$$p(s_1, s_2|t, y) = \frac{p(t|s_1, s_2)p(s_1, s_2)}{p(t)} = \frac{p(s, t)}{p(t)} \tag{8}$$

$$p(s, t) = \mathcal{N}(\begin{bmatrix} s \\ t \end{bmatrix}; \begin{bmatrix} \mu_s \\ \mu_t \end{bmatrix}, \begin{bmatrix} \Sigma_s & 0 \\ 0 & \Sigma_t|s \end{bmatrix}) \tag{9}$$

so $p(s_1, s_2|t, y)$ can be written as:

$$\begin{aligned} p(s|t) &= \mathcal{N}(s; \mu_{s|t}, \Sigma_{s|t}) \\ \Sigma_{s|t} &= (\Sigma_s^{-1} + A^T \Sigma_{t|s}^{-1} A)^{-1} \\ \mu_{s|t} &= \Sigma_{s|t}(\Sigma_s^{-1}\mu_s + A^T \Sigma_{t|s}^{-1}(t - b)) \\ A &= \begin{bmatrix} 1 & -1 \end{bmatrix} \end{aligned} \tag{10}$$

3.2 $p(t|s_1, s_2, y)$

According to question

$$\begin{aligned} p(t|s_1, s_2, y) &\propto p(y|t)p(t|s_1, s_2) \\ p(y|t) &= \text{sign}(t) \end{aligned} \tag{11}$$

we have

$$p(t|s_1, s_2, y) = \mathcal{N}(t; s_1 - s_2, \sigma_t^2) \tag{12}$$

3.3 $p(y = 1)$

From the question

$$p(y = 1) = p(t > 0) \tag{13}$$

t is obtained by marginalizing out s_1 and s_2 from $p(t, s_1, s_2)$ Using equations (7) and (8)

$$\begin{aligned} p(t) &= N(t; \mu_t, \Sigma_t) \\ \mu_t &= A\mu_s + b \\ \Sigma_t &= \Sigma_{t|s} + A\Sigma_s A^T \end{aligned} \tag{14}$$

4 Gibbs Sampler

4.1

First, we set both s_1 and s_2 have the mean = 1
The graph of posterior distributions with samples = 1000 have mean = 1.528 and 0.586

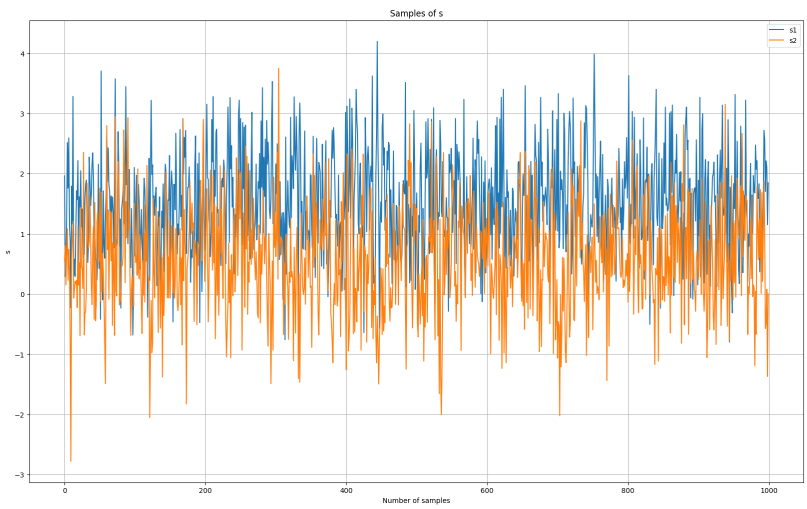

Figure 2: Gibbs sampler with samples = 1000, burn-in = 0

To determine a reasonable burn-in, we draw a graph to see the mean of samples from 0 to 500.

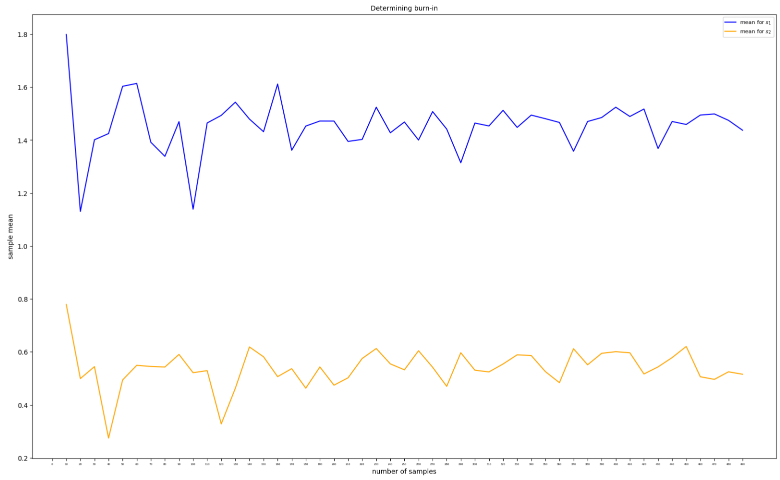

Figure 3: Determine burn-in

Apparently, samples within 0 - 50 have the most fluctuation mean so we choose burn-in = 50. Then re-run the experiment:

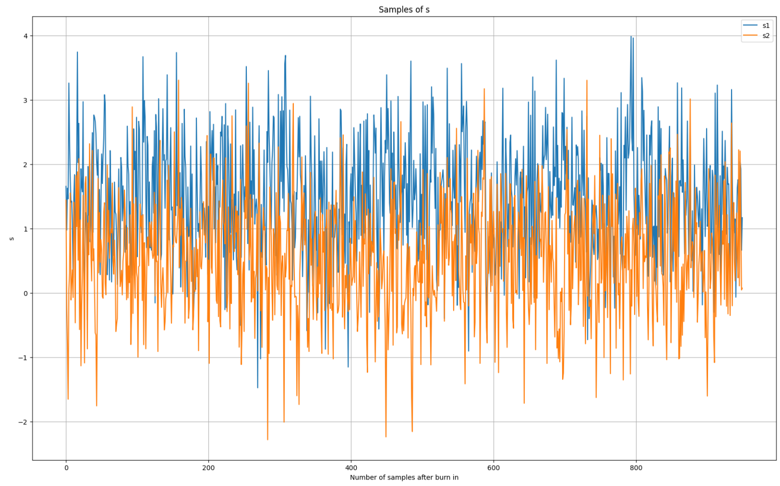

Figure 4: Gibbs sampler with samples = 1000, burn-in = 50

This time, the mean = 1.483 and 0.541, and we achieved a better experiment.

4.2

The function that can transform the samples drawn from the Gibbs sampler into Gaussian distributions can be seen in the attachment *Final.ipynb*.

4.3 & 4.4

We integrate these two questions into one graph, and the time required to draw the samples is labeled in every title of the subplot.
According to the graph, as samples increases, the time required increases significantly, and as the histogram shows, samples = 1000 is a good choice.

Through the comparison of the prior and posterior, we can know that player 1 won and player 2 lost, since player 1 has a positive skill level while player 2 has a negative skill level.

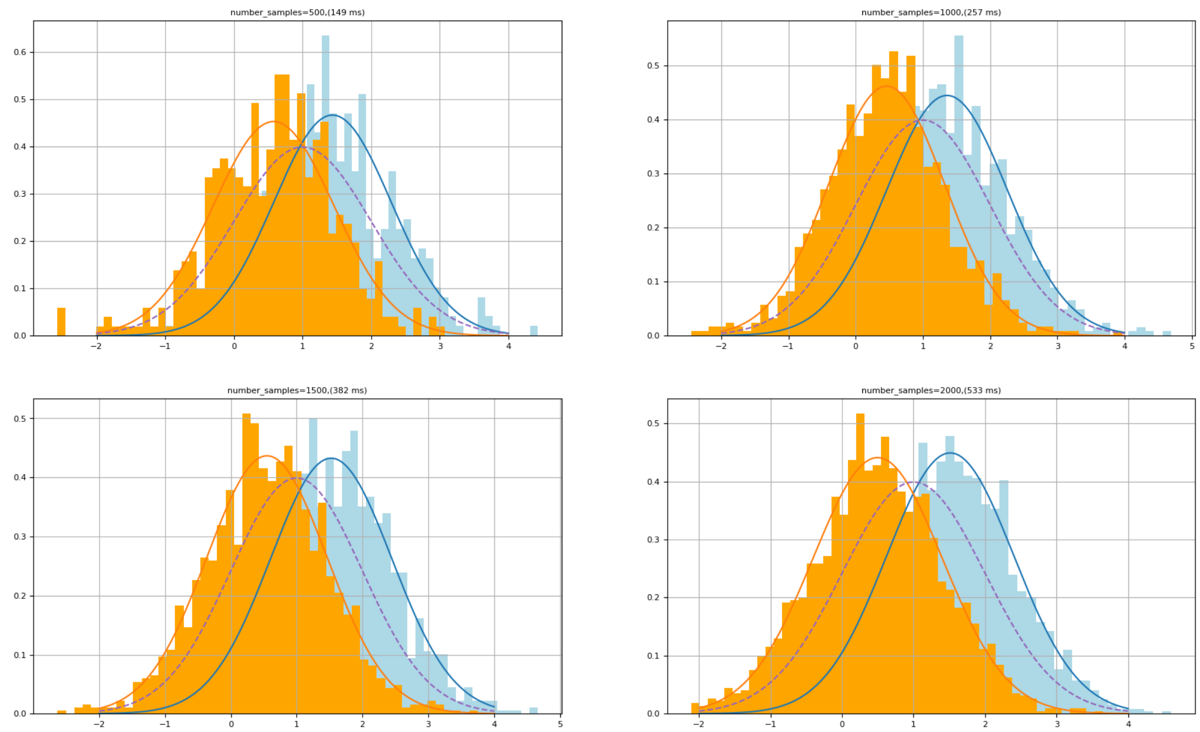

Figure 5: Compare prior with the Gaussian approximation of posterior

5

The final ranking list is (a), and after changing the order of the matches, the ranking list is (b). The variance can be interpreted as how stable a team's performance is when in competition, or the uncertainty to win the game.

The outcome varies depending on the order of the matches because if a team with a low skill score wins against a team with a high skill score, this low-score team's mean and variance will increase and decrease respectively, while the high-score team's mean and variance will change in the opposite trend. When such changes accumulate, they can make some changes in the final ranking.

	mean	var	Rank
Juventus	1.974026	0.053944	1
Torino	1.817804	0.050080	2
Napoli	1.588347	0.056420	3
Milan	1.544789	0.058040	4
Atalanta	1.460297	0.053437	5
Inter	1.403475	0.050428	6
Roma	1.395993	0.049273	7
Sampdoria	1.221448	0.043531	8
Lazio	1.009116	0.075034	9
Bologna	0.859866	0.045571	10
Fiorentina	0.843589	0.045353	11
Udinese	0.756451	0.050837	12
Cagliari	0.753335	0.050182	13
Empoli	0.749141	0.032008	14
Genoa	0.744783	0.046191	15
Sassuolo	0.693322	0.046033	16
Parma	0.671735	0.046419	17
Spal	0.520815	0.044364	18
Frosinone	0.227761	0.054595	19
Chievo	0.207966	0.061919	20

(a) rank list

	mean	var	Rank
Juventus	1.949164	0.049147	1
Torino	1.904036	0.058243	2
Napoli	1.722595	0.040653	3
Roma	1.642179	0.048037	4
Milan	1.584182	0.044248	5
Atalanta	1.433728	0.052546	6
Inter	1.417802	0.037925	7
Sampdoria	1.135131	0.044947	8
Lazio	1.095210	0.039774	9
Bologna	0.953989	0.051815	10
Fiorentina	0.871598	0.040809	11
Udinese	0.834159	0.063390	12
Cagliari	0.823141	0.047176	13
Parma	0.748261	0.046541	14
Sassuolo	0.724832	0.061413	15
Empoli	0.717570	0.054993	16
Genoa	0.704471	0.042869	17
Spal	0.674124	0.042054	18
Frosinone	0.278805	0.041672	19
Chievo	0.188100	0.059542	20

(b) rank list with shuffled order

6 Using the model for predictions

The accordance code can be seen in the attachment *Final.ipynb*, and the prediction rate is 0.761.

7 Factor graph

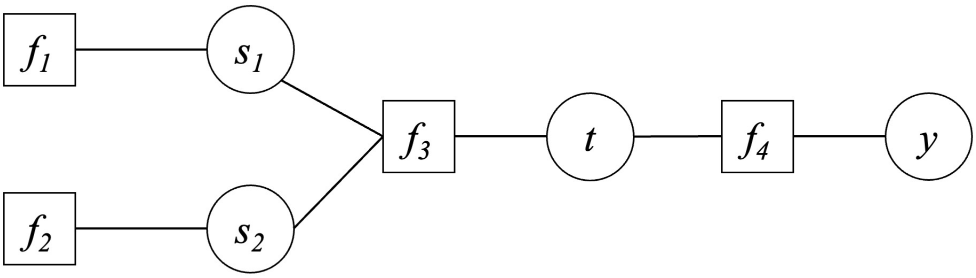

Figure 7: Factor Graph

4 variable nodes: s_1, s_2, t, y
4 factor nodes

$$
\begin{aligned}
f_1 &= \mathcal{N}(s_1; \mu_{s_1}, \Sigma_{s_1}) \\
f_2 &= \mathcal{N}(s_2; \mu_{s_2}, \Sigma_{s_2}) \\
f_3 &= \mathcal{N}(t; s_1 - s_2, \sigma_t^2) \\
f_4 &= \text{sign}(t)
\end{aligned} \tag{15}
$$

8 A message-passing algorithm

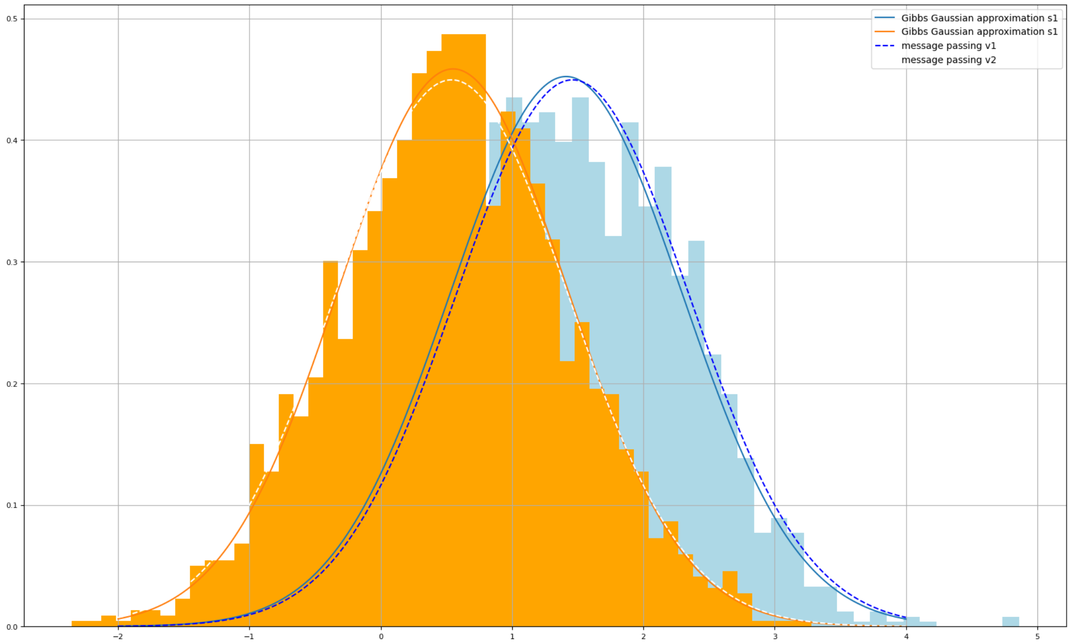

Figure 8: posteriors computed with message-passing vs.Gaussian approximation from Gibbs Sampling

9 Your own data

We use the dataset of *NBA Basic Player Data by Game* games 2010s, and the rank list can be seen in the attached ipynb document, the correspondence prediction rate is 0.687.

10 Open-ended project extension

In this part, we improved the prediction algorithm to get a better prediction rate 0.858 while the previous is 0.761.