CUED - Engineering Tripos Part IIB 2019-2020 **Module Coursework**

Module	4F13	Title of report	Coursework Two - Probabilistic Ranking
Date submitted:	22/11/2019	Assessment for this module is ☒ 100% / ☐ 25% coursework of which this assignment forms _____ %	

UNDERGRADUATE STUDENTS ONLY		POST GRADUATE STUDENTS ONLY			
Candidate number:	5584F	Name:		College:	

Feedback to the student

☐ **See also comments in the text**

		Very good	**Good**	Needs improvmt
CONTENT	**Completeness, quantity of content:** Has the report covered all aspects of the lab? Has the analysis been carried out thoroughly?			
	Correctness, quality of content Is the data correct? Is the analysis of the data correct? Are the conclusions correct?			
	Depth of understanding, quality of discussion Does the report show a good technical understanding? Have all the relevant conclusions been drawn?			
	Comments:			
PRESENTATION	**Attention to detail, typesetting and typographical errors** Is the report free of typographical errors? Are the figures/tables/references presented professionally?			
	Comments:			

Overall assessment (circle grade)		A*	**A**	B	C	D
Guideline standard		>75%	**65-75%**	55-65%	40-55%	<40%
Penalty for lateness:		*20% of marks per week or part week that the work is late.*				

Marker: Date:

4F13 - Coursework Two - Probabilistic Ranking

Candidate Number: 5584F

Word count: 989

December 6, 2019

I. a

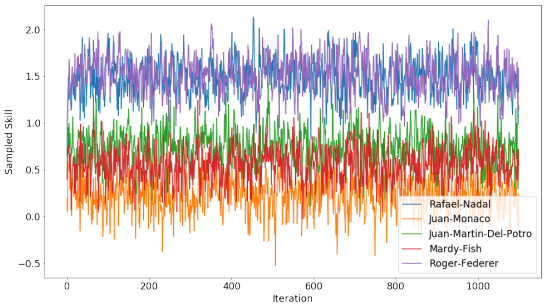

Figure 1: *Sampled skills from Gibbs sampling*

Using the data available for tennis matches played and their outcome, the ranking of each players skill and chance of beating another opponent can be calculated. One method of doing this is to use Gibbs sampling. Gibbs sampling is used to jointly sample player skills by sampling from simpler conditional distributions instead of the joint. Figure 1 shows these samples being random and different for each player. This method of computing sequential samples leads to dependence between samples. Therefore a burn in time is required after initialising. The burn in is required because the initial variable values may not reside in locating of high probability in the joint distribution, this means several iterations of Gibbs sampling are required for the samples to converge on the desired distribution. Convergence happens when the mean and standard deviation settle which can be found by computing these terms against iteration number. Figure 3 shows this convergence, a burn in time of 200 iterations was used in subsequent sampling. Figure 2 shows the

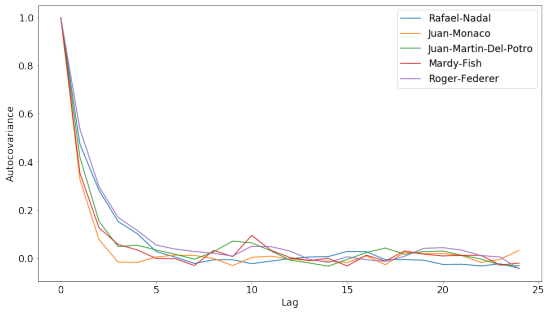

Figure 2: *Autocovariance of Gibbs sampling*

autocovariance of sampled skills compared to their lag. in order to get independent samples this autocovariance should be 0. A lag of 5 was chosen for sampling independently in later tests. Due to the burn in time and only using 1/5 of all samples the Gibbs sampling algorithm should be run for over 1000 iterations to get enough samples that can be reliably used for ranking.

II. b

Another method of evaluating marginal skill distributions is Expectation Propagation (EP) using message passing. This process involves updating the marginal skill distributions after each instance of propagating the distribution though the factor graphs. Each iteration fits the marginal distributions better to the available game data.

The marginal distributions are initialised with

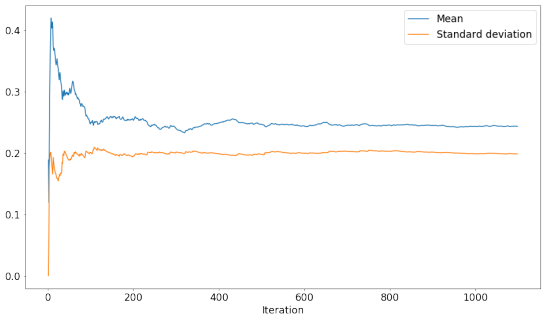

Figure 3: *Convergence to a distribution*

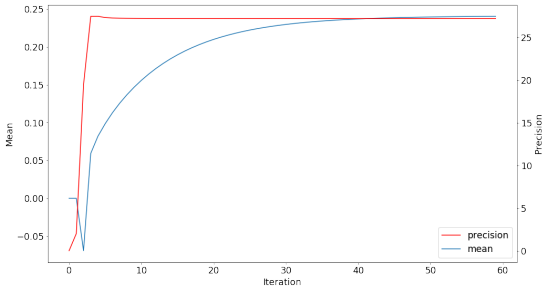

Figure 4: *Convergence of distribution using message passing*

a mean and variance, as the message passing iterates these will converge to the desired distribution of skills. Figure 4 shows this convergence for one player, the precision settles very quickly within 5 iterations unlike the mean which takes 40 iterations. The number of iterations required will be different for each player as the difference between desired distribution and initial distribution varies. This convergence to a distribution is the same as in Gibbs sampling where the samples converge upon a certain mean and variance.

III. C

The top four players on the ATP ranking are Novak Djokovic, Rafael Nadal, Roger Federer and Andy Murray the probability of a players skill and performance being higher then another can be computed using the distributions from EP. The is done by calculating the probability p($skill_1 - skill_2$>0), and treating $skill_1 - skill_2$ as the random variable. Tables

	ND	RN	RF	AM
ND	0.5	0.060	0.091	0.0147
RN	0.940	0.5	0.573	0.233
RF	0.909	0.427	0.5	0.189
AM	0.985	0.767	0.811	0.5

Table 1: *Probability column player has higher skill than row player, using distribution from EP*

	ND	RN	RF	AM
ND	0.5	0.345	0.362	0.280
RN	0.655	0.5	0.518	0.427
RF	0.638	0.482	0.5	0.409
AM	0.720	0.573	0.591	0.5

Table 2: *Probability column player has higher performance than row player, using distribution from EP*

1 and 2 list these probabilities for skill and game performance respectively. The performance corresponds to the chance a player wins a given game, this is due to the skill difference plus noise. Figure 5 shows these distributions between two players. The performance distribution is much wider due to the added uncertainty which results in the less skilled player increasing their chance to win. This is also seen when comparing the probabilities in tables 1 and 2.

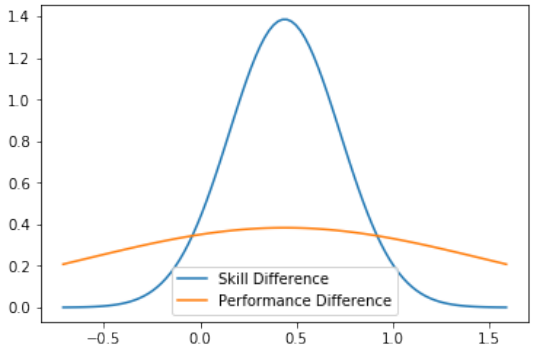

Figure 5: *Difference between skill and performance for Nadal and Djokovic*

IV. D

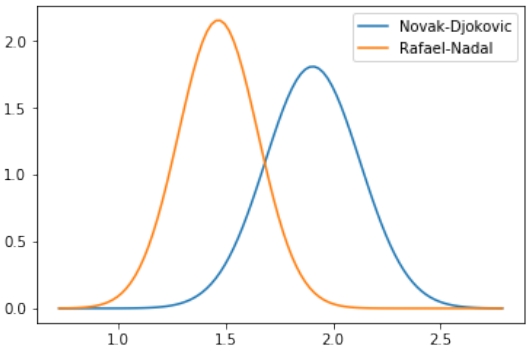

Figure 6: *Marginal skill distributions with means* $[1.905,\ 1.466]^T$ *and standard deviations* $[0.220,\ 0.185\ ]^T$

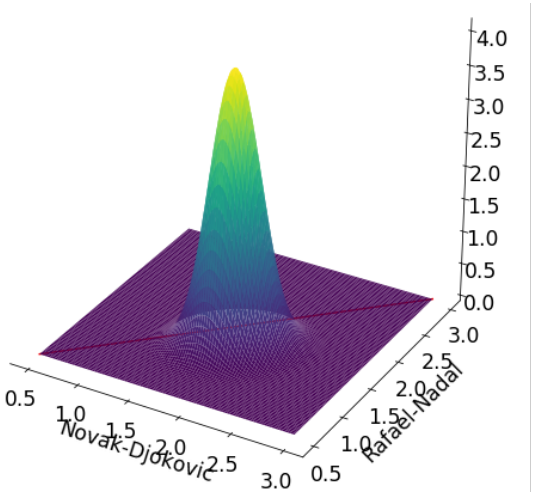

Figure 7: *Joint skill distribution with means* $[1.905,\ 1.466]^T$, *variances* $[0.0486,\ 0.0343]^T$ *and symmetric covariance 0.0096*

Method	p($Djokovic_s > Nadal_s$)
Marginal distributions	0.936
Joint distribution	0.959
From samples	0.972

Table 3: *Probability Djokovic has high skill than Nadal computed using three methods*

The skills of two players can be compared using various methods when using Gibbs sampling. The first method is to use the skill marginals, which can be formed by taking the mean and variance of the skill samples and approximating each distribution as normal. The two marginal Gaussians shown in figure 6 can then be used to determine the probabilities as before. The Gibbs samples also contain information about the covariance between players, once calculated along with the mean a join distribution can be formed. This is once again approximated to be Gaussian, with the probabilities now calculated from the volume either side of the line $skill_1 = skill_2$ as in figures 7 and 8. The final method utilises the samples directly, by comparing the skills at a given iteration of the algorithm while ensuring burn in and autocorrelation times are accounted for. Table 3 shows the probability that Djokovic is more skilled then Nadal for each method, all resulting in similar probabilities.

V. D

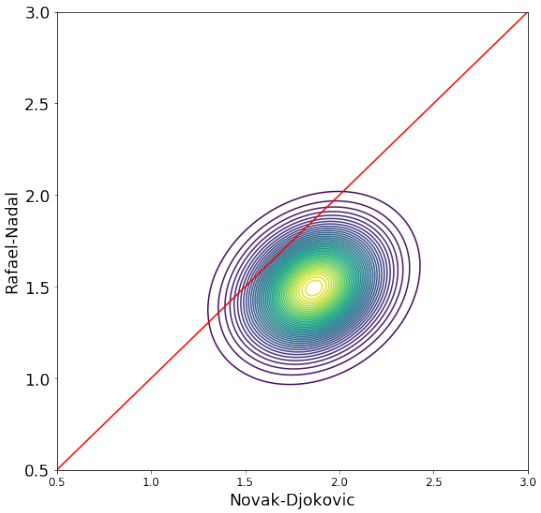

Figure 8: *Joint skill distribution contour plot*

Each method has various strengths and downfalls. The marginal distribution and using the samples directly is computationally faster then using a 2D joint Gaussian. However, just using marginal distributions ignores the extra information in the covariance that the other methods take into account, this information contains player pairing specific information.

Using the samples directly can result in very noisy probabilities unless lots of samples are used. This means that if the distribution can be reasonably assumed to be Gaussian its best to use the joint distribution as this considers covariance and smooths sample noise.

	ND	RN	RF	AM
ND	0.5	0.040	0.072	0.006
RN	0.960	0.5	0.579	0.204
RF	0.928	0.421	0.5	0.168
AM	0.994	0.796	0.832	0.5

Table 4: *Probability column player has higher skill than row player, using joint distribution*

Table 4 contains the probabilities one player is more skilled then another using the joint Gaussian distribution. Comparing this to table 1 from messaging passing its clear that both methods produce very similar results as all probabilities differ by 0.03 at most. This means the distributions generated with each method are also similar.

VI. E

The ranking of each player can be created using the mean skills of each player from both the message passing and Gibbs sampling. These can be seen in figures 10 and 11 respectively. These both produce the same ranking, further showing that the converged distributions in each method are similar. Another method of creating a ranking is to calculate the win percentage for each player using the game data. This doesn't take into account how skilled each opponent was in each game, meaning losing to bad players counts as much as losing to good players. Figure 9 shows this list. This list depicts a similar overall position for each player, but with lots of the order shuffled slightly meaning player position can differ by around 8 places compared to the other methods. This method is clearly inferior to Gibbs and EP however it is computational faster for generating an approximate list.

VII. Appendix - Code listings

i. a

```
m[p] = sum(t[np.where(G[:, 0] == p)]) -
    ↪  sum(t[np.where(G[:, 1] == p)])
       for g in range(N):
           iS[G[g, 0], G[g, 0]] += 1
           iS[G[g, 1], G[g, 1]] += 1
           iS[G[g, 0], G[g, 1]] -= 1
           iS[G[g, 1], G[g, 0]] -= 1
```

ii. c

```
def prob_skill_higher(p1,p2,means,pres,
    ↪ performance=0):
    #probability that s1>s2 = p(s1-s2
        ↪ >0) let z=s1-s2 , z~(m1-m2,
        ↪ v1+v2)
    p1=int(p1)
    p2=int(p2)
    mean = means[p1,-1]-means[p2,-1]
    var = 1/pres[p1,-1] + 1/pres[p2,-1]
        ↪  + performance
    prob = 1 - norm.cdf((0-mean)/(var
        ↪ **0.5))
    return prob
```

iii. d

```
for i in range(2):
    data[i,:] = skill_samples[int(ids[i
        ↪ ]),:][burn_in:-1]
    means[i]=np.mean(data[i,:])


cov = np.cov(data)
```

```
autocov =5
count=0
burn_in=200
it= int((skill_samples.shape[1]-burn_in-1)//autocov )


for i in range(it):
    x1 = skill_samples[int(ids[0]),burn_in + autocov*i ]
    x2 = skill_samples[int(ids[1]),burn_in + autocov*i ]
    if x1>x2:
        count+=1
prob = count/it
```

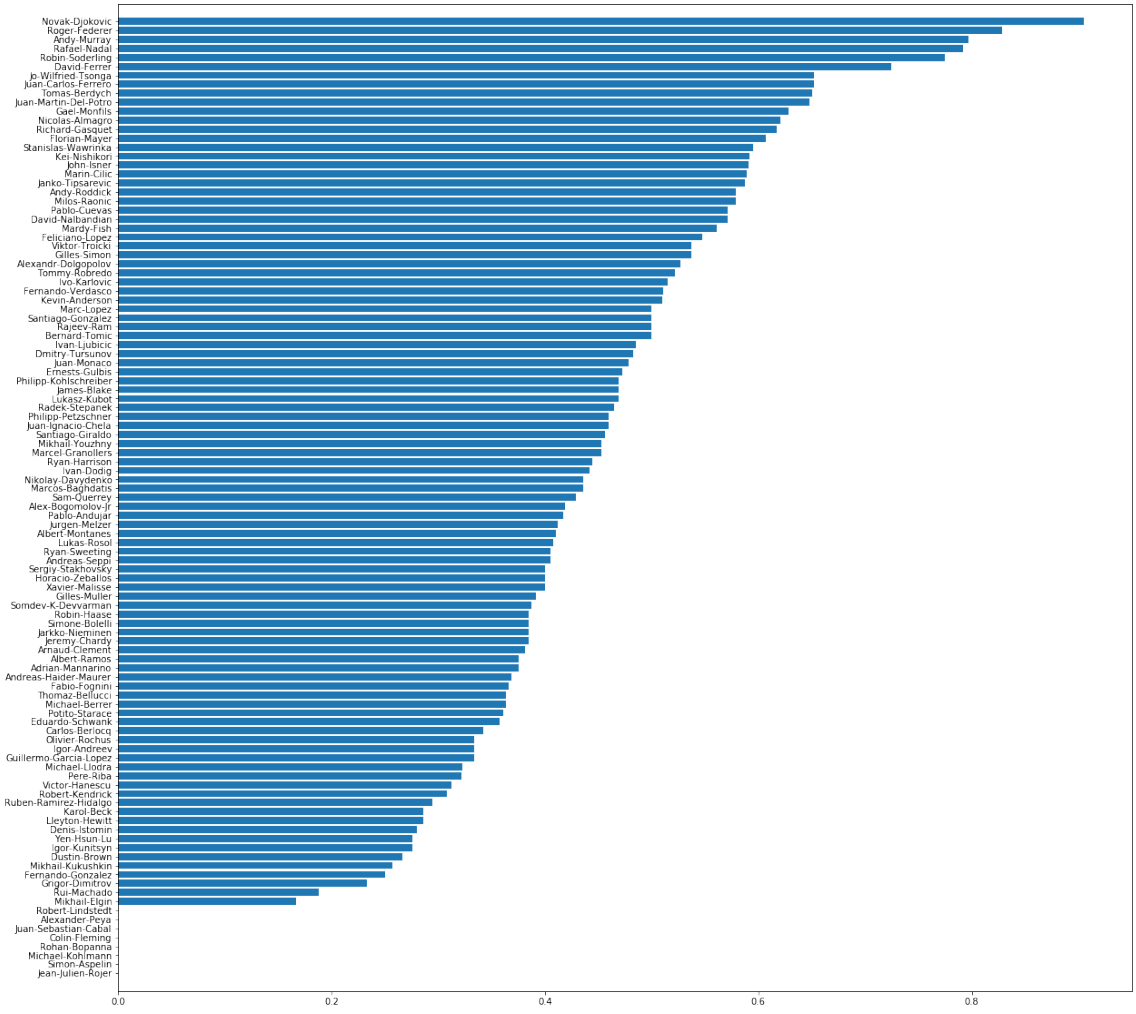

Figure 9: *Player order using empirical results*

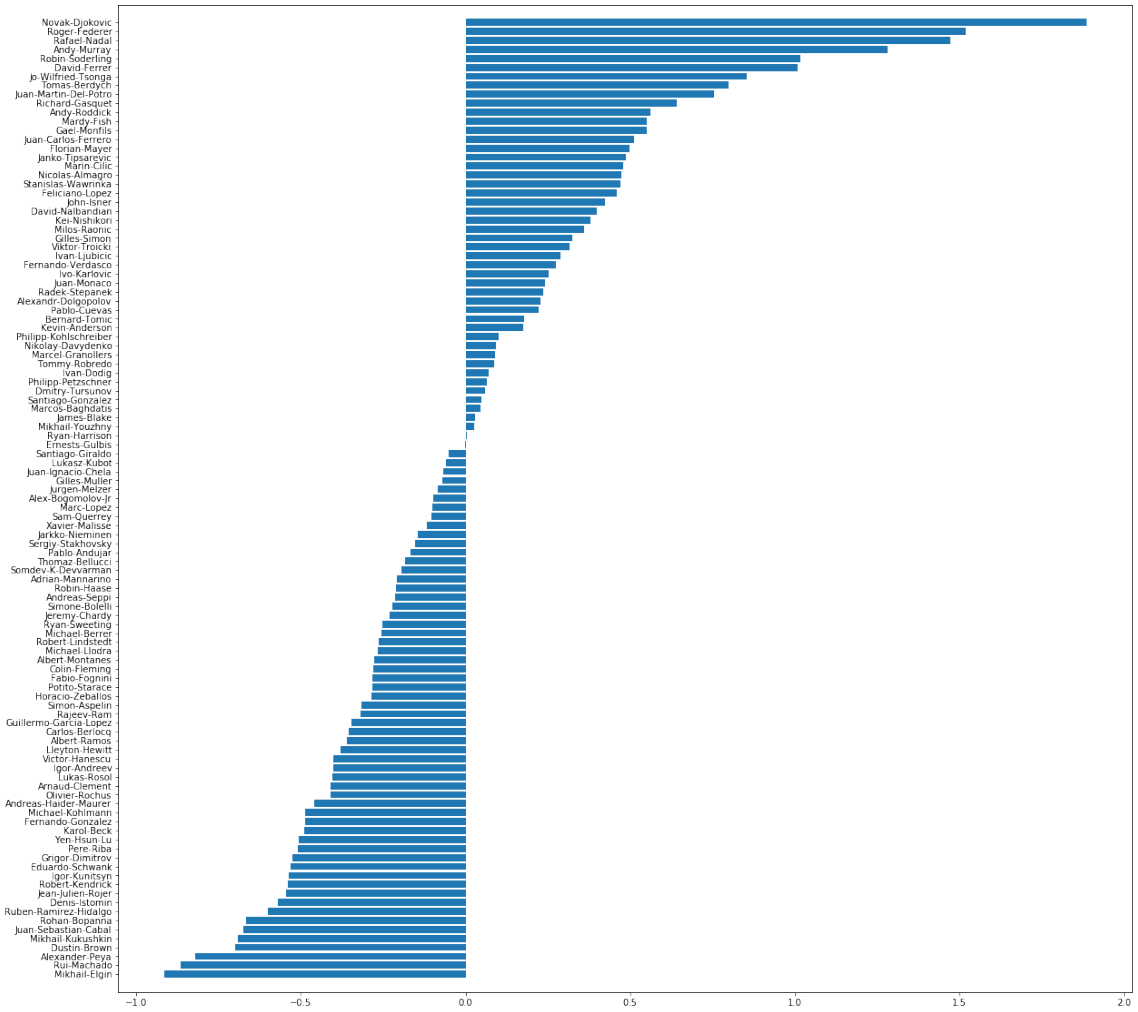

Figure 10: *Player order using Gibbs sampling*

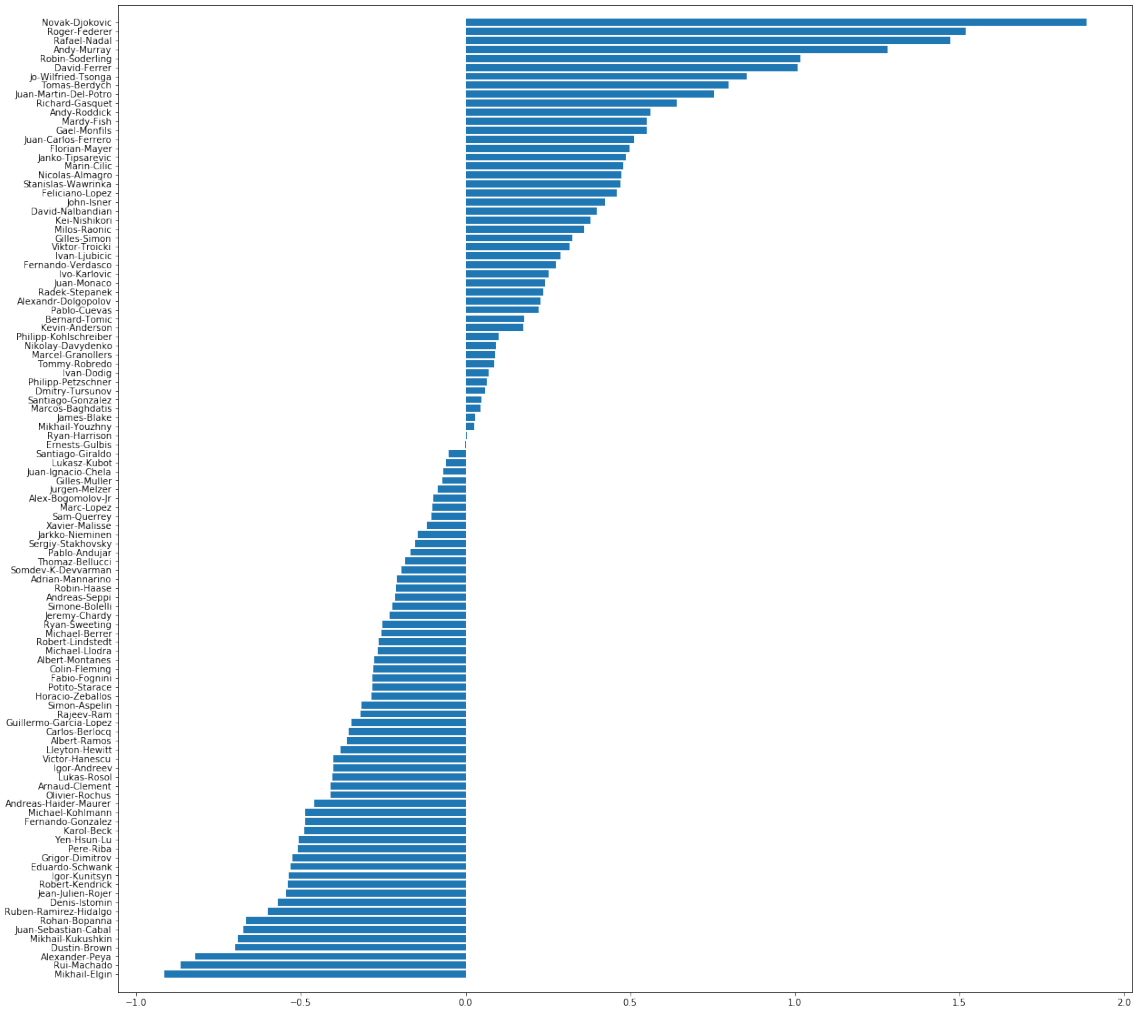

Figure 11: *Player order using message passing*