BAYESIAN RATING MODELS

STARCRAFT II

Introduction to Bayesian Statistics, Korea University

Adam Frederik Ingwersen Linnemann, 2018952670

31. 12. 2018

Abstract

This report investigates the power of bayesian rating models in predicting professional Starcraft II league matches. Prediction on outcomes for the 2017 Korean championships are being considered; a variety of distributional relations are proposed, but it is found that the modelling of attacking and defensive parameters are suitable - and should follow a normal distribution. The proposed model rightly predicts the winner of the 2017 championships - but only some of the runner-ups are predicted correctly. More complicated models are proposed to alleviate this drawback of the base model.

Contents

	match_date	player_1	player_1_match_status	score	player_2	player_2_match_status	player_1_race	player_2_race	addon	tournament_type
0	09/19/2016	MC	[loser]	0–2	Stats	[winner]	P	P	LotV	online
1	09/19/2016	MC	[winner]	2–1	NaTuRal	[loser]	P	T	LotV	online
2	09/19/2016	MC	[loser]	1–2	Dark	[winner]	P	Z	LotV	online
3	09/13/2016	MC	[loser]	0–2	INnoVation	[winner]	P	T	LotV	online

Figure 1: Raw data

1 Scope and Purpose

The aim of this report is to investigate the performance and interpretability of bayesian rating models, as proposed by Baio & Blangiardo (2009), on Starcraft II match data.

2 Data

Data on Starcraft II matches from 2014 to 2017 is obtained from Kaggle. The data contains 187.397 match results played between 1.856 distinct players (professional and non-professional).

In order to achieve a sensible amount of data for MCMC simulation on a laptop, the data has to be subsetted. This is done by looking at match-dates after *2016-01-01*. Furthermore, the number of players are limited to 8, by grabbing the top 8 players in the **2017 Starcraft II World Championship** - where the top 8 constitutes the finalists of that tournament.

```
xpath = "//tr/td[@align = 'left']/a/text()"
url = "https://liquipedia.net/starcraft2/2017_StarCraft_II_World_Championship_Series_Korea/
page = requests.get(url)
tree = html.fromstring(page.content)
top  = tree.xpath(xpath)
top8 = top[:8]
```

Further data preprocessing steps are documented in the attached `report.ipynb`.
The players' defensive and offensive performance is represented in the figure below for the two considered seasons.

In addition, the proportionality should ideally be somewhat equivalent between players. It's apparent that **Rogue** and **TY** are slightly underrepresented in the data.

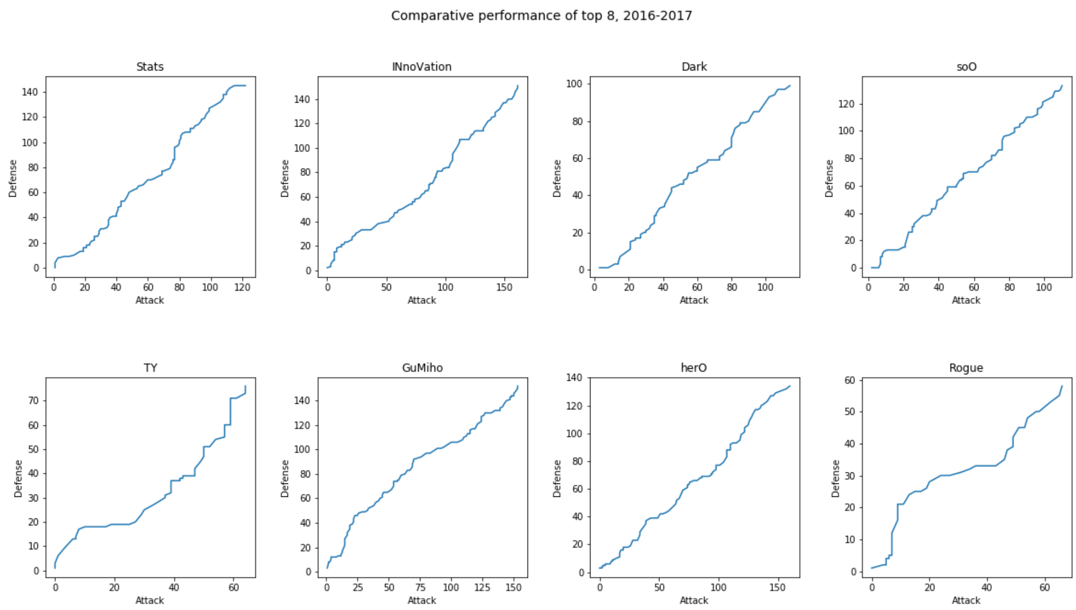

Figure 2: Relative player def/off performace

Table 1: Player proportion of total dataset

players	proportion
Stats	0.1497731
INnoVation	0.1527988
Dark	0.1180030
soO	0.1240545
TY	0.0816944
GuMiho	0.1543116
herO	0.1512859
Rogue	0.0680787

3 Modelling

In attempting to model the relations present in a Starcraft II tournament and capture the inhrerent dynamics that apply to such competitions, three distinct models are proposed. Each of which are to be explained in sections below. All models rely on having a set of proxy-distributions that represent the offensive and defensive attributes of any particular player. The model statistics and related plots are presented in the attached `ipython-notebook`.

3.1 Baseline Model

This model serves as the basline to which other models are evaluated. It's a stripped down version of the model presented by *Baio & Blangiardo (2010)*, that takes a fixed zero-mean in offensive/defensive attributes, and only the parameter variance is modeled explicitly.

3.1.1 Specification

```
tau_att    = pm.Gamma('tau_att', 0.1, 0.1)
tau_def    = pm.Gamma('tau_def', 0.1, 0.1)
atts_star = pm.Normal("atts_star", mu = 0, sd = tau_att, shape = n_players)
defs_star = pm.Normal("defs_star", mu = 0, sd = tau_def, shape = n_players)
atts = pm.Deterministic('atts', atts_star - tt.mean(atts_star))
defs = pm.Deterministic('defs', defs_star - tt.mean(defs_star))
p1_theta = tt.exp(atts[player_1] + defs[player_2])
p2_theta = tt.exp(atts[player_2] + defs[player_1])
p1_points = pm.Poisson('p1_points', mu = p1_theta, observed = observed_player1_score)
p2_points = pm.Poisson('p2_points', mu = p2_theta, observed = observed_player2_score)
```

3.1.2 Results

With NUTS-sampling, the model converges quickly and achieves excellent **Rubin-Gelman** statistics and **BFMI (Bayesian Fraction of Missing Information)**, which indicates that the model both converged and is well-specified, respectively.

3.2 Intercept Model

This model is a re-write of the model presented in *Baio & Blangiardo (2010)*, with a tunable intercept to indicate a home-court advantage to one of the teams. This intuitively should not provide any additional information in a Starcraft II context.

3.2.1 Specification

```
tau_att    = pm.Gamma('tau_att', 0.1, 0.1)
tau_def    = pm.Gamma('tau_def', 0.1, 0.1)
intercept  = pm.Flat('intercept')
atts_star = pm.Normal("atts_star", mu = 0, sd = tau_att, shape = n_players)
defs_star = pm.Normal("defs_star", mu = 0, sd = tau_def, shape = n_players)
atts = pm.Deterministic('atts', atts_star - tt.mean(atts_star))
defs = pm.Deterministic('defs', defs_star - tt.mean(defs_star))
p1_theta = tt.exp(intercept + atts[player_1] + defs[player_2])
p2_theta = tt.exp(intercept + atts[player_2] + defs[player_1])
p1_points = pm.Poisson('p1_points', mu = p1_theta, observed = observed_player1_score)
p2_points = pm.Poisson('p2_points', mu = p2_theta, observed = observed_player2_score)
```

3.2.2 Results

This specification has been run several times under the exact same conditions as the baseline model, but does not appear to converge - and may be misspecified as well. This suggests, that the inclusion of the intercept as a trainable parameter only makes MCMC more complicated, and does not provide any additional information. Thus, the homecourt-advantage is not present in the observed Starcraft II mathces, and perheaps in Starcraft II in general.

3.3 Tunable Mean Model wo. Intercept

This model is, again, inspired by the one suggested by *Baio & Blangiardo (2010)*, but disregards the intercept, and uses a trainable mean for both the offensive and defensive stats of the players. This seems to be a more sensible approach in the Starcraft II context.

3.3.1 Specification

```
mu_att     = pm.Normal('mu_att', 0)
mu_def     = pm.Normal('mu_def', 0)
tau_att    = pm.Gamma('tau_att', 0.1, 0.1)
```

```python
tau_def    = pm.Gamma('tau_def', 0.1, 0.1)
atts_star = pm.Normal("atts_star", mu = mu_att, sd = tau_att, shape = n_players)
defs_star = pm.Normal("defs_star", mu = mu_def, sd = tau_def, shape = n_players)
atts = pm.Deterministic('atts', atts_star - tt.mean(atts_star))
defs = pm.Deterministic('defs', defs_star - tt.mean(defs_star))
p1_theta = tt.exp(atts[player_1] + defs[player_2])
p2_theta = tt.exp(atts[player_2] + defs[player_1])
p1_points = pm.Poisson('p1_points', mu = p1_theta, observed = observed_player1_score)
p2_points = pm.Poisson('p2_points', mu = p2_theta, observed = observed_player2_score)
```

3.3.2 Results

This model performs rather similar to the baseline model, but does yield a lot of divergences in the sampling process, and takes longer to converge - thus it should be discarded in favor of the baseline model.

4 Discussion of major findings

The first major finding has been to keep the model simple, as there does not appear to be complicated interactions in the data, that the model has a hard time capturing. Had the data been more detailed or more scarce, it may have been that the baseline model would not suffice.

4.1 Picking the best model

The best results are, surprisingly, obtained by the baseline model, which shows telltale signs of convergence and proper specification.

4.2 Starcraft v. Soccer

Obviously Starcraft II is a very different sport from soccer. The fact, that soccer is played in teams, whilst Starcraft II is played by individuals at the competitive level is a good example of the differences pertaining to the sports. As elaborated above, the does not appear to be any homecourt advantage, which is a rather significant factor in determining match outcomes. The

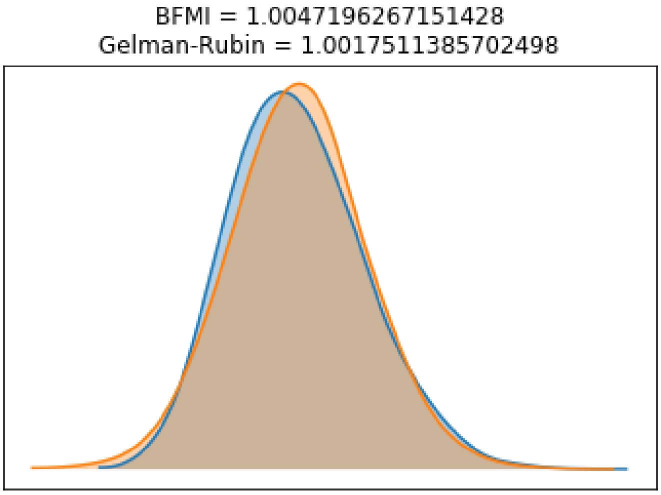

Figure 3: Specification and convergence of baseline-model

proposed modelling framework assumes, that the offensive and defensive properties are seperable in Starcraft II, which may not be the case in the same way as in Soccer. It may very well be the case, that one should include the player activity such as APM (Actions Per Minute) as a factor. In Starcraft II defensive strategies also often rely of offensive harrasment of the opponent, which the proposed modelling framework may have trouble picking up on.

With that being said, it is not unreasonable to think that there is a tradeoff for players attempting offensive strategies by eroding defensive capabilites in the short term.

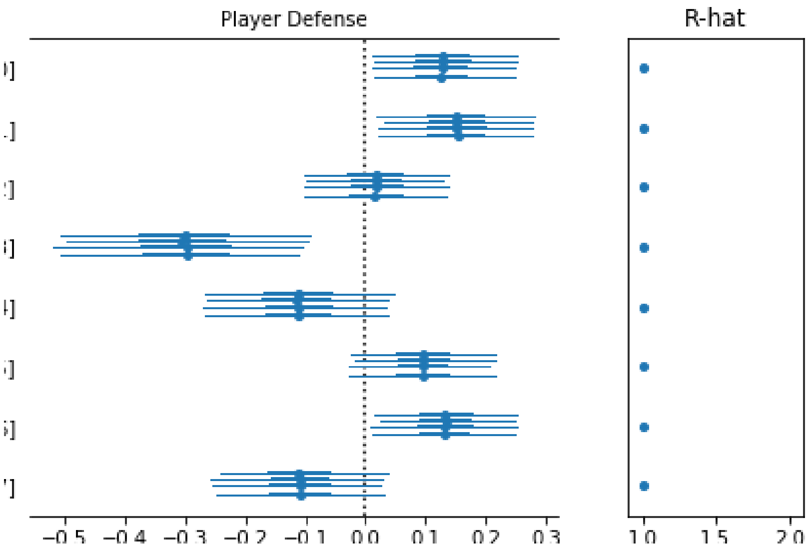

Figure 4: Player Defensive Capabilites

The defensive capabilites are directly linked to the opponents score - so the lower defensive

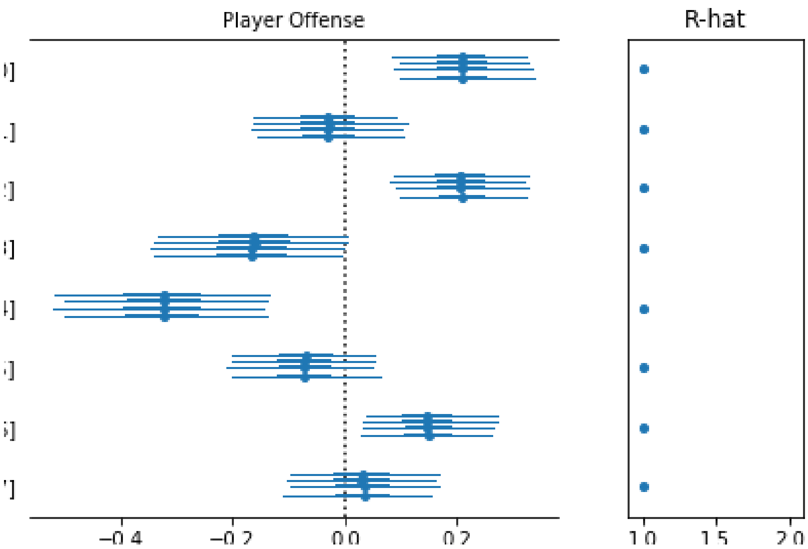

Figure 5: Player Offensive Capabilites

capabilites, the better the player is at not letting the opponent get points.

5 Implications

The work that has been conducted this far can be trivially used to resample from the model and thus predict the final ranking of the players in the tournament.

rank player	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0
Dark	0.066	0.178	0.406	0.686	1.151	0.503	0.010	0.000
GuMiho	0.469	0.755	0.787	0.589	0.335	0.065	0.000	0.000
INnoVation	0.271	0.492	0.769	0.790	0.506	0.170	0.002	0.000
Rogue	0.000	0.000	0.000	0.000	0.000	0.000	0.144	2.856
Stats	1.621	0.769	0.374	0.166	0.054	0.016	0.000	0.000
TY	0.000	0.000	0.000	0.001	0.010	0.142	2.759	0.088
herO	0.817	0.923	0.634	0.444	0.145	0.037	0.000	0.000
soO	0.004	0.022	0.072	0.230	0.699	1.884	0.089	0.000

Figure 6: 2017 Tournament Predictions

We can compare to the actual final ranking:

Player Details					GSL			Other						
#			ID	WCS Points	S1	S2	S3							
1			Stats	12875	4000	1200	1900	1050	2100	675	0	1500	225	225
2			INnoVation	11050	1200	600	4000	1500	900	1500	225	0	675	450
3			Dark	7950	600	800	1900	675	1350	0	450	1050	675	450
4			soO	7750	2800	2800	800	450			0		450	450
5			TY	7500	1200	1200	1200		3000		225			675
6			GuMiho	6800	200	4000	800		900		450		225	225
7			herO	6775	1200	800	800	225	150		1500	0	1050	1050
8			Rogue	5600	200	1200	1200						1500	1500

Figure 7: 2017 Tournament Results

6 Concluding Remarks

This project has shown that bayesian rating models certainly are applicable to predicting Starcraft II rankings, although the methodology applied falls a bit short.

Future projects should seek to explore binomial distributions for the offensive and defensive capabilites and focus on a smaller player pool.