

Hierarchical Bayesian Modeling of the English Premier League

Milad Kharratzadeh

09 January, 2017

Contents

Introduction	2
Model	2
Reading and Munging the Data	3
Stan Code	4
Fitting the Model	5
Evolution of Team Abilities	6

Introduction

In this case study, we provide a hierarchical Bayesian model for the English Premier League in the season of 2015/2016. The league consists of 20 teams and each two teams play two games with each other (home and away games). So, in total, there are 38 weeks, and 380 games. We model the score difference (home team goals – away team goals) in each match. The main parameters of the model are the teams’ abilities which is assumed to vary over the course of the 38 weeks. The initial abilities are determined by performance in the previous season plus some variation. Please see the next section for more details.

We implement and fit our model in **Stan** and prepare the data and analyze the results in **R**.

Model

The score difference in game i , denoted as y_i , is modeled as a t distribution:

$$y_i \sim t_\nu(a_{home_week(i), home_team(i)} - a_{away_week(i), away_team(i)} + b_{home}, \sigma_y),$$

where $a_{w,j}$ is the ability of team j in week w . Because of the irregularities in the schedule of the games, the ‘week’ for the home and away teams may not be the same; so, $home_week(i)$ and $away_week(i)$ denote the week for the home and away team in game i respectively. The possible advantage (or disadvantage) for the home team is modeled by the variable b_{home} ; we do not expect this effect to be large, and therefore, assign it a $N(0, 10)$ weak prior. The variation in the score difference is modeled by σ_y and we give it a weak prior $N(0, 10)$. The degrees of freedom, ν , has a prior of $\text{Gamma}(2, 0.1)$ ¹.

We assume that the abilities of the teams can change during the season (due to injuries, player forms, etc.). We assume the following random walk model:

$$a_{w,j} \sim N(a_{w-1,j}, \sigma_{aj}), \quad w = 2, \dots, 38$$

where σ_{aj} models the game-to-game variability for team j . We assume a hierarchical structure where the variations in team abilities are sampled from $N(0, \tau_a)$, and $\tau_a \sim \text{Cauchy}(0, 1)$.

The ability for the first week is derived from the previous season performance with some variability:

$$a_{1,j} \sim N(b_{prev}a_{0,j}, \sigma_{a0}),$$

where b_{prev} is the regression coefficient and $a_{0,j}$ is a score between -1 and 1 achieved by a linear transformation of the total points achieved from last season. We expect some team-level variation in the initial performance; this is modeled by σ_{a0} . Both b_{prev} and σ_{a0} have weakly informative priors, $N(0, 10)$.

We fit our model every week using all the matches up until that point. (Therefore, we fit our model 38 times.)

¹As suggested by: Juarez and Steel, “Model-based clustering of non-Gaussian panel data based on skew-t distributions”, *Journal of Business & Economic Statistics* 28 (2010), 52-66

Reading and Munging the Data

We first read the data from the website `football-data.co.uk` and save it to a list called `epl`. The main components of this list are `home_team`, `away_team`, and `score_diff` which have 380 elements each. Teams have fixed IDs which are integers from 1 to 20 assigned to teams sorted alphabetically. The previous performance (points in previous season) is stored a separate CSV file; this data is read and mapped to a score between -1 and $+1$ using the user-defined function `map_to_score`. The variable `home_week`, also of length 380, identifies the ‘week’ for the home team (i.e., how many matches the home team has played so far, including the current match).

```
library(plyr)
# Linear map of points to a score between 0 and 1
map_to_score <- function(x) {
  x_max <- max(x);
  return(x/x_max);
}
url_csv <- paste("http://www.football-data.co.uk/mmz4281/1516/E0.csv",
                 sep="") # Data downloaded from football-data.co.uk
mydat <- read.csv(url(url_csv)); epl <- c();
# teams are assigned IDs 1, 2, ...:
epl$home_team <- as.numeric(mydat$HomeTeam)
epl$away_team <- as.numeric(mydat$AwayTeam)
epl$team_names <- levels(mydat$HomeTeam)
epl$home_goals <- mydat$FTHG # FTHG: full time home goals
epl$away_goals <- mydat$FTAG # FTHG: full time away goals
epl$score_diff <- epl$home_goals - epl$away_goals
# Points from last season are read and mapped to a score
epl$prev_perf <- read.csv('DATA/prev_perf.csv', header = FALSE)
epl$prev_perf <- map_to_score(epl$prev_perf[,2])
epl$nteams <- length(unique(epl$home_team))
epl$ngames <- length(epl$score_diff)
epl$nweeks <- floor(2*epl$ngames/epl$nteams)
# The following code computes the week for each team in their games:
epl$home_week <- c(); epl$away_week <- c();
for (g in 1:epl$ngames) {
  epl$home_week[g] <- sum(epl$home_team[1:g] == epl$home_team[g]) +
    sum(epl$away_team[1:g] == epl$home_team[g])
  epl$away_week[g] <- sum(epl$away_team[1:g] == epl$away_team[g]) +
    sum(epl$home_team[1:g] == epl$away_team[g])
}
epl$bet_home <- mydat$B365H; # Betting odds for home team win
epl$bet_draw <- mydat$B365D; # Betting odds for draw
epl$bet_away <- mydat$B365A; # Betting odds for away team win
saveRDS(epl, 'epl_data.rds')
```

Stan Code

The Stan code for the model is shown below. The code is commented and self-explanatory. In the `generated quantities` block, we sample replications data for the `score_diff`; we will use these later for posterior predictive checks.

```
functions {
  real skellam_lpmf(int y, real mu_1, real mu_2) {
    return bessel_first_kind(abs(y), 2*sqrt(mu_1*mu_2)) * ((mu_1/mu_2)^(y*0.5)) * exp(-m
  }
}

data {
  int<lower=1> nteams; // number of teams (20)
  int<lower=1> ngames; // number of games
  int<lower=1> nweeks; // number of weeks
  int<lower=1> home_week[ngames]; // week number for the home team
  int<lower=1> away_week[ngames]; // week number for the away team
  int<lower=1, upper=nteam> home_team[ngames]; // home team ID (1, ..., 20)
  int<lower=1, upper=nteam> away_team[ngames]; // away team ID (1, ..., 20)
  int score_diff[ngames]; // home_goals - away_goals
  row_vector<lower=0>[nteam> prev_perf; // a score between 0 and +1
}

parameters {
  real b_home; // the effect of hosting the game in mean of score_diff dist.
  real b_prev; // regression coefficient of prev_perf
  real<lower=0> sigma_a0; // teams ability variation
  real<lower=0> tau_a; // hyper-param for game-to-game variation
  real<lower=0> sigma_y; // score_diff variation
  row_vector<lower=0>[nteam> sigma_a_raw; // game-to-game variation
  matrix[nweeks,nteam> eta_a; // random component
}

transformed parameters {
  matrix<lower=0>[nweeks, nteam> a; // team abilities
  row_vector<lower=0>[nteam> sigma_a; // game-to-game variation
  a[1] = exp(b_prev * log(prev_perf) + sigma_a0 * eta_a[1]); // initial abilities (at we
  sigma_a = tau_a * sigma_a_raw;
  for (w in 2:nweeks) {
    a[w] = exp(log(a[w-1]) + sigma_a .* eta_a[w]); // evolution of abilities
  }
}

model {
```

```

// Priors
b_prev ~ normal(0,1);
sigma_a0 ~ normal(0,1);
sigma_y ~ normal(0,1);
b_home ~ normal(0,1);
sigma_a_raw ~ normal(0,1);
tau_a ~ cauchy(0,1);
to_vector(eta_a) ~ normal(0,1);
// Likelihood
for (g in 1:ngames) {
  score_diff[g] ~ skellam(a[home_week[g],home_team[g]], a[away_week[g],away_team[g]]);
}
}

```

Fitting the Model

As mentioned earlier, we fit the model multiple times, after every 10 games. In the code below, `epl_w` contains all the data for matches in the first $w \times 10$ matches. We fit the model with 4 chains of length 750 (with the first half for warmup), for a total of 1500 samples (after warmup).

```

library("rstan")
epl <- readRDS("epl_data.rds")
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
sm <- stan_model("epl_model_skellam.stan")
nsamples <- 1500
a_sims <- array(NA, c(nsamples, epl$nweeks, epl$ntteams))
for (w in 1:38) {
  epl_w <- epl
  idx <- c(1:(w*10))
  epl_w$home_team <- epl$home_team[idx]
  epl_w$away_team <- epl$away_team[idx]
  epl_w$home_goals <- epl$home_goals[idx]
  epl_w$away_goals <- epl$away_goals[idx]
  epl_w$score_diff <- epl$score_diff[idx]
  epl_w$home_week <- epl$home_week[idx]
  epl_w$away_week <- epl$away_week[idx]
  epl_w$ngames <- w*10
  epl_w$nweeks <- max(c(epl_w$home_week, epl_w$away_week))
}

```

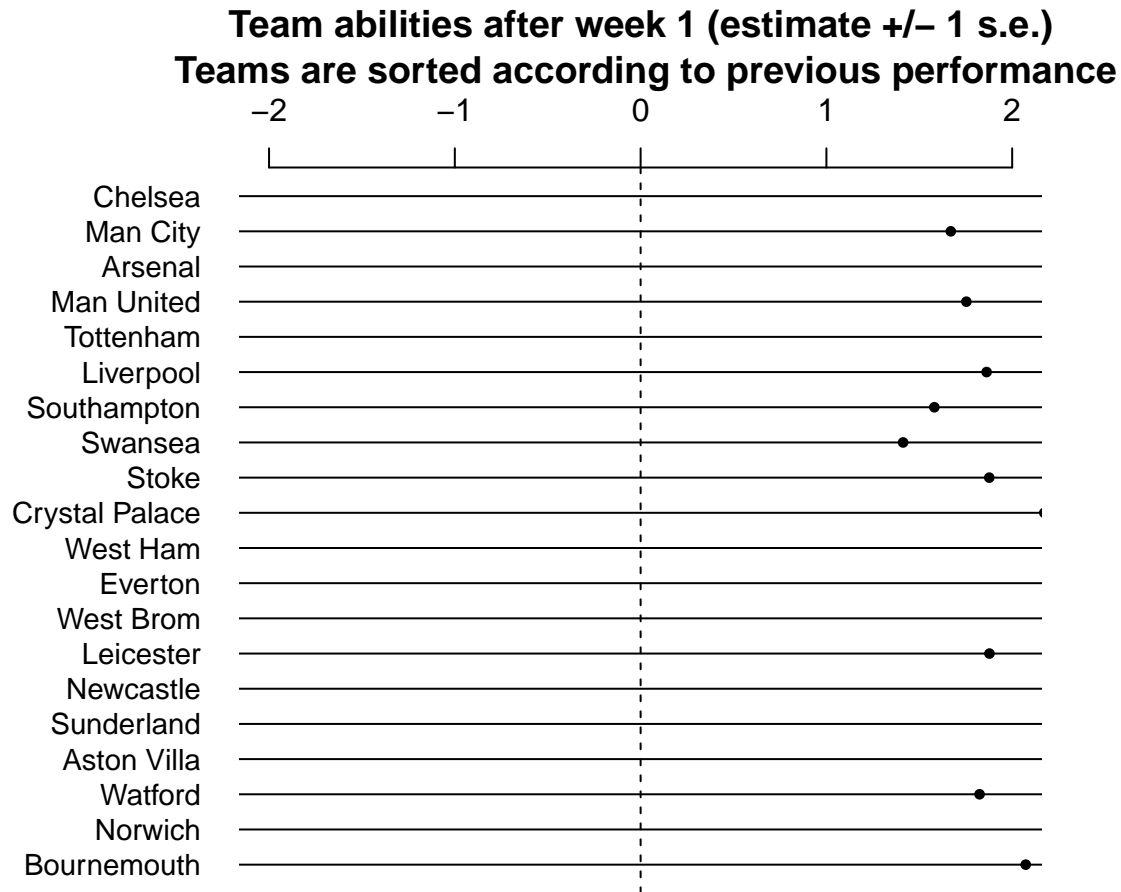
```

fit <- sampling(sm, chains = 4, iter = (nsamples/2), data = epl_w)
saveRDS(fit, paste("FITS/fit_", w, ".rds", sep=""))
sims <- extract(fit)
for (g in ((w-1)*10 + 1):(w*10)) {
  a_sims[,epl$home_week[g],epl$home_team[g]] <-
    sims$a[,epl$home_week[g],epl$home_team[g]]
  a_sims[,epl$away_week[g],epl$away_team[g]] <-
    sims$a[,epl$away_week[g],epl$away_team[g]]
}
}
saveRDS(a_sims,"FITS/a_sims.rds")

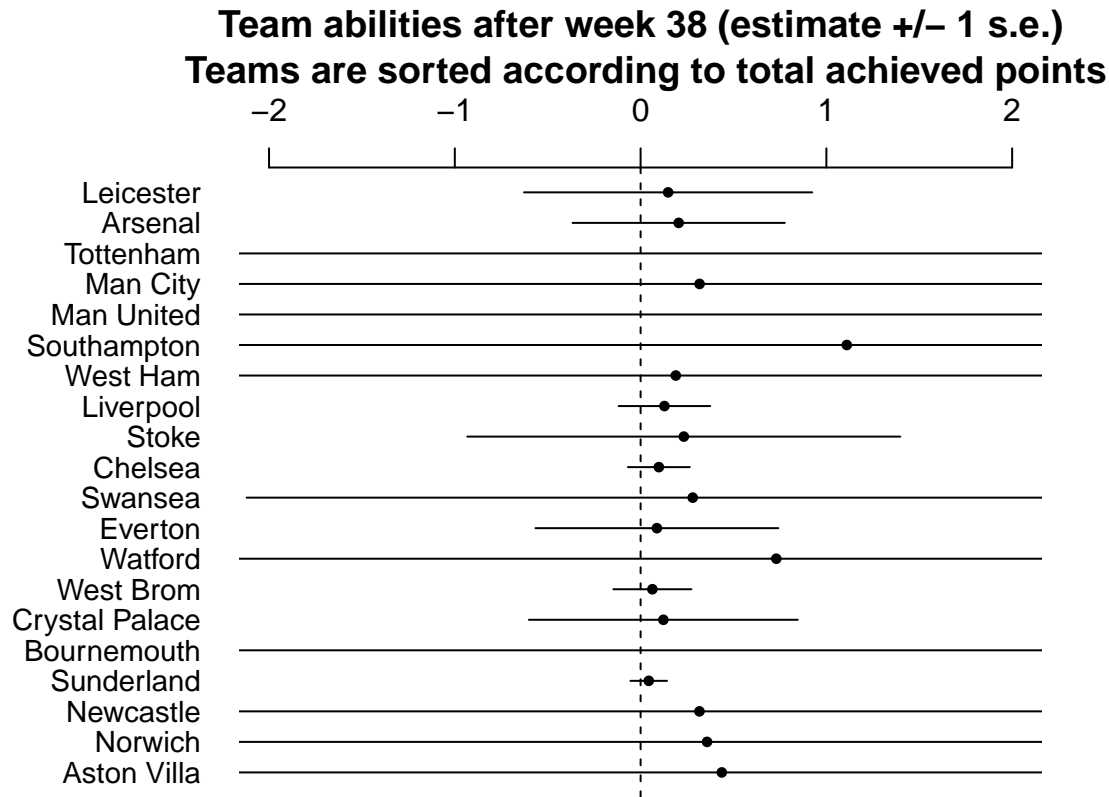
```

Evolution of Team Abilities

Remember that we re-fit the model after each week. In the first table below, we show the estimated team abilities (± 1 s.e.) after week 1. The teams in the table are sorted according to their performance in the previous season. We observe that after only one match for each team, the estimated abilities are somewhat similar to previous performance with perturbations due to the results in the first week. We also observe that the uncertainty intervals are quite wide; this also makes sense, because we have only one observation per team.



In the next figure, we plot the estimated abilities at the end of the season (i.e., after week 38). This time, we sort the teams in the tables according to their final standings in the table (i.e., sorted according to total points) at the end of the season. We observe that the estimated abilities are fairly consistent with the actual rankings and the uncertainty intervals are narrower compared to the results after week 1.



We also examine the evolution of abilities for each team over the course of the season. In Fig. 2, we plot the estimated abilities for all the teams after each week (using the data from matches upto and including that week). The uncertainty intervals (\pm 1 s.e.) are shown with gray bands. The score differences are shown by red dots. We observe that ability movements are consistent with score differences. For instance, Leicester did not do well in the previous season; so, its initial ability is not high. Despite some good results at the beginning of the season, they got some bad results in weeks 4 to 9 (a drop in the estimated ability). Afterwards, they did very well until the end of the season which is shown by the positive trend in the estimated abilities.

Matchday	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Ground	H	A	H	A	H	A	H	A	A	H	A	H	A	H	A	H	A	A	H	H	A	A	H	H	A	A	H	H	A	H	A	H	A	H	H	A	H	A
Result	W	W	D	D	W	D	L	W	D	W	W	W	W	D	W	W	W	L	D	D	W	D	W	W	W	L	W	D	W	W	W	W	D	W	D	W	D	
Position	1	1	1	3	2	3	6	4	5	5	3	3	1	2	1	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 1: Leicester performance and position in the table (source: Wikipedia)

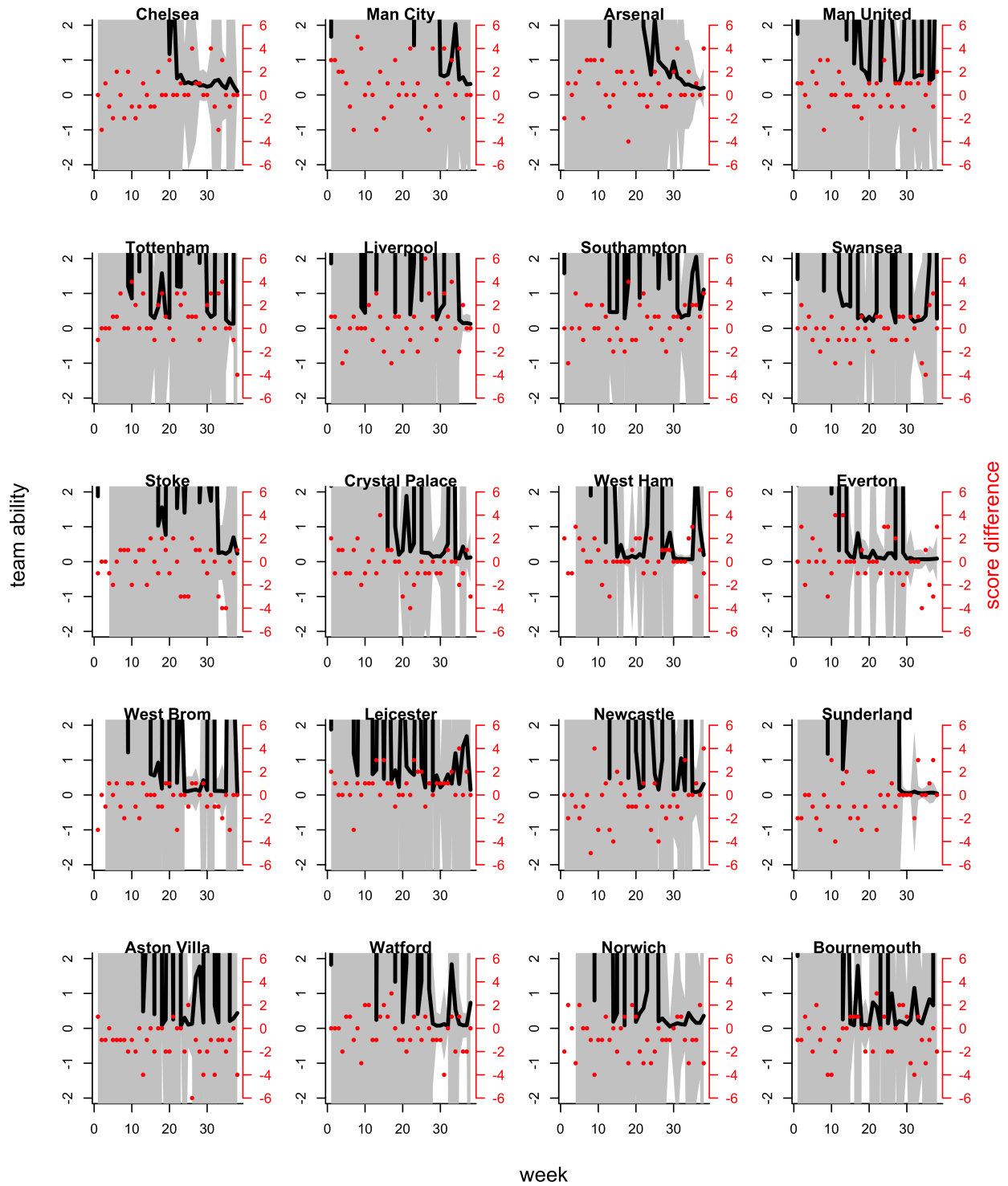


Figure 2: For each of the 20 teams in the division, the black line shows estimated ability as the season goes on (at each point, posterior mean ± 1 s.e. given only the first w weeks of the season, for each w); the red dots display the score difference for each game.