Analysis of Serie A 2009-10 Results

Ourfali Marc

17 janvier 2020

Introduction

- Goal : Predict Results from 2009-2010.
- Models used :
  1. Poisson distribution
  2. Skellam distribution
  3. Zero inflated version of Skellam's distribution (ZPD)
- Methods implemented :
  1. Metropolis-Within-Gibbs
  2. Metropolis-Adjusted Langeving (MALA)
  3. Gradient descent

Dataset

- Data set : Seasons 2006-2007 till 2009-2010 from Serie A. We have dropped the teams that where relegated before season 2009-2010.

index	Home	Away	HG	AG	Result
0	Fiorentina	Inter	2	3	A
1	Roma	Livorno	2	0	H
3	Cagliari	Catania	0	1	A
4	Chievo	Siena	1	2	A
6	Milan	Lazio	2	1	H

Table – Head of the Data set

- In 2009-10 We have the following Results over 380 matches :
  1. Home Win H = 186
  2. Draw D =102
  3. Away Win A=92

Updated Coefficients

- For each teams give an attacking and defensive coefficient.
- Coefficient Δ gives the Home Advantage.

Team	Att	Dif
Atalanta	0.010361	0.002564
Bari	-0.001984	0.002543
Bologna	-0.003227	0.011944
Cagliari	0.005107	0.015559
Catania	0.003522	-0.011179

TABLE – Head of coefficient Data frame

- $\Delta, Att_i, Att_j, Dif_i, Dif_j \sim \mathcal{N}(0, 0.1) \quad i, j \in \{2, \ldots, 20\}$
- $Att_1 = -\sum_{i=2}^{20} Att_i$ et $Dif_1 = -\sum_{i=2}^{20} Dif_i \Rightarrow$ identifiable model

Poisson Model

- Predict the number of Home and Away goals of team i and j respectively for each match.
- Home Goals are given by $HG_{i,j} \sim \mathcal{P}(\lambda_{HG_{i,j}})$
- Away Goals are given by $AG_{i,j} \sim \mathcal{P}(\lambda_{AG_{i,j}})$
- $\lambda_{HG_{i,j}} = exp(\Delta + Att_i - Dif_j)$
- $\lambda_{AG_{i,j}} = exp(Att_j - Dif_i)$

Implementation

- Parameters : $\theta = (\Delta, Att_2, \ldots, Att_{20}, Dif_2, \ldots, Dif_{20})$
- Prior : $f_p(\theta) = f_{norm}(\Delta) \times \prod_{i=2}^{20} f_{norm}(Att_i) \times \prod_{i=2}^{20} f_{norm}(Dif_i)$

  $f_{norm}(x) = \frac{1}{\sqrt{2\pi}0.1} e^{\frac{1}{2}\frac{x}{0.1^2}}$
- Likelihood : $f_L(D|\theta) = \prod_{i,j=2}^{20} f_{poiss}(HG_{i,j}, \lambda_{HG_{i,j}}) \times f_{poiss}(AG_{i,j}, \lambda_{AG_{i,j}})$

  $f_{poiss}(k, \lambda) = e^{-\lambda} \frac{k^{\lambda}}{k!}$

Skellam Model

- Predict the difference of the goal scored between the two teams rather than the number of home and away goals.
- Model goal difference as Skellam distribution : probability distribution of the difference of two statistically independent random Poisson-distributed variables.
- Main Idea.Incorporate correlation between the goals in a football game.
  $\Rightarrow$ Empirical evidence has shown relatively low correlation.
- Difference given by : $SD_{i,j} = HG_{i,j} - AG_{i,j} \sim PD(\lambda_{HG_{i,j}}, \lambda_{AG_{i,j}})$, with $\lambda_{HG_{i,j}}$ and $\lambda_{AG_{i,j}}$ defined as before.

Implementation

- Parameters : $\theta = (\Delta, Att_2, \ldots, Att_{20}, Dif_2, \ldots, Dif_{20})$
- Prior : $f_p(\theta) = f_{norm}(\Delta) \times \prod_{i=2}^{20} f_{norm}(Att_i) \times \prod_{i=2}^{20} f_{norm}(Dif_i)$

  $f_{norm}(x) = \frac{1}{\sqrt{2\pi}0.1} e^{\frac{1}{2}\frac{x}{0.1^2}}$
- Likelihood : $f_L(D|\theta) = \prod_{i,j=2}^{20} f_{Skellam}(S_{i,j}, \lambda_{H_{i,j}}, \lambda_{A_{i,j}})$

  $f_{Skellam}(k, \lambda_1, \lambda_2) = e^{-(\lambda_1+\lambda_2)} \left(\frac{\lambda_1}{\lambda_2}\right)^k I_{|k|}(2\sqrt{\lambda_1\lambda_2})$

  where $I_r(x) = \left(\frac{x}{2}\right) \cdot \sum_{k=0}^{\infty} \frac{\left(\frac{x^2}{4}\right)^k}{k!\Gamma(r+k+1)}$

ZPD Model

- Zero Inflated version of our previous Skellam Model.
  $\Rightarrow$ Allows for frequent zero-valued observations.
- This new prior permits to control the excess of draws adding a new parameter p.
- Parameter p initialized as $p \sim \mathcal{U}(0,1)$.
- For the predictions generate a random variable $a \sim \mathcal{U}(0,1)$.
  - If $a < p$: $SD_{i,j} = 0$,
  - Else : $SD_{i,j} \sim PD(\lambda_{HG_{i,j}}, \lambda_{AG_{i,j}})$

Implementation

- Parameters : $\theta = (\Delta, Att_2, \ldots, Att_{20}, Dif_2, \ldots, Dif_{20}, p)$
- Prior : $f(\theta) = f_{norm}(\Delta) \times \prod_{i=2}^{20} f_{norm}(Att_i) \times \prod_{i=2}^{20} f_{norm}(Dif_i)$

  $f_{norm}(x) = \frac{1}{\sqrt{2\pi}0.1} e^{\frac{1}{2}\frac{x}{0.1^2}}$
- Likelihood : $f(D|\theta) = \prod_{i,j=2}^{20} f_{ZPD}(S_{i,j}, \lambda_{H_{i,j}}, \lambda_{A_{i,j}}, p)$

$$f_{ZPD} = \begin{cases} p + (1-p) f_{Skellam}(0, \lambda_1, \lambda_2) & \text{if} \quad k = 0 \\ (1-p) f_{Skellam}(k, \lambda_1, \lambda_2) & \text{if} \quad k \neq 0 \end{cases}$$

Metropolis within Gibbs

Algorithm :

Given : target : $\widetilde{f}$, proposal :$q(\eta)$

Input : θ_0, N_{max}

for i in 1 : N_{max} :

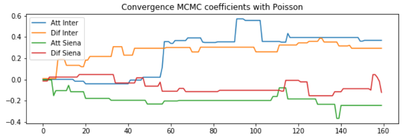
1. sample a proposal : $\theta^c_{i+1} \sim q(.,\theta_i)$
2. Compute M-H acceptance ratio : $\alpha = \min\{\frac{\widetilde{f}(\theta^c|D)}{\widetilde{f}(\theta|D)}, 1\}$
3. $u \sim \mathcal{U}nif(0,1)$
   If $u < \alpha$: $\theta_{i+1} = \theta^c_{i+1}$
   Else $\theta_{i+1} = \theta_i$

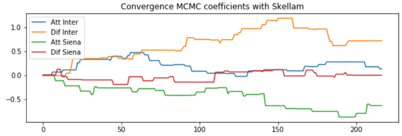
Output : Mean of the accepted θ^c

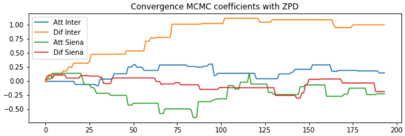
Implemented :

- $\theta_0 \sim \mathcal{N}(0, 0.1)^d$
- $q(\theta_i, \theta^c_{i+1}) = q(\theta^c_{i+1}, \theta_i)$
  At each iteration update 5 random coefficients adding δ_i to each of them :
  $\delta_i \sim \mathcal{N}(0, 0.1), \ i = \{1, \cdots, 5\}$
- $\widetilde{f}(\theta|D) = f_p(\theta) \times f_L(D|\theta)$
- N $=$ 10 000

Convergence of the Model







- Acceptance ratio : around 20% for each model.
- Coefficient in accord with the number of goals scored and conceded by each team.

Match Prediction

- For each match calculate $\lambda_{HG_{i,j}}$ and $\lambda_{AG_{i,j}}$.
- Skellam and Poisson Model :1000 simulations of the posterior distribution are generated.
- Average the goals and set draw if the difference between the two results $\in (0; 0.25)$.
  $\Rightarrow$ Limit the number of draws to get better predictive performance.
- ZPD Model : Generate only one simulation of the posterior distribution.

Results

- For each model obtain a high Home Advantage coefficient : $\Delta \approx 0.35$.

- Poisson Model.
- 50.03% of well predicted results.
- Distribution :
  - H = 246
  - D = 57
  - A= 77
- Computationally faster ; Small results obtain.

- Skellam Model.
- 49.47% of well predicted results.
- Distribution :
  - H = 223
  - D = 55
  - A=102
- Computationally slower.

- ZPD Model.
- 48.94% of well predicted results.
- Distribution :
  - H = 278
  - D = 68
  - A=36
- Here $p \to$ 0.05.
- p is independent of the teams.

Conclusion Metropolis Within Gibbs Model

- Advantage : Method easily implemented with satisfying results.
- Model easily incorporate new prior information.
- Easy to interpret : used by bookmakers.
- Drawbacks : Need to choose adapted δ_i to add to the coefficients.
- For Skellam and ZPD Models gives to much weights to the defense coefficients.
- Model : Seems best to choose Poisson Model.

Metropolis-Adjusted Langeving (MALA)

Algorithm :

Given : target : $\widetilde{f}$, proposal :$q(\eta)$

Input : θ_0, N_{max}, τ

for i in 1 : N_{max} :

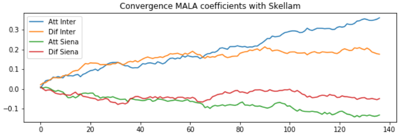
1. Calculate gradient : $\nabla(log(f(\theta_i|D)))$
2. Proposal $\theta^c_{i+1} = \theta_i + \tau \times \nabla(log(f(\theta_i|D))) + \sqrt{2\tau}\epsilon$
3. Compute $\alpha = \min\{\frac{\widetilde{f}(\theta^c_{i+1}|D)}{\widetilde{f}(\theta_i)|D)} \frac{q(\theta_i,\theta^c_{i+1})}{q(\theta^c_{i+1},\theta_i)}, 1\}$
4. $u \sim \mathcal{U}nif(0,1)$
   If $u < \alpha$: $\theta_{i+1} = \theta^c_{i+1}$
   Else $\theta_{i+1} = \theta_i$

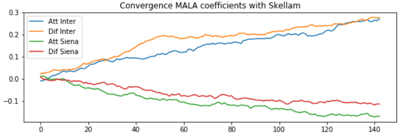
Output : Mean of the accepted θ^c

Implemented :

- $\theta_0 \sim \mathcal{N}(0, 0.1)^d$
- $\widetilde{f}(\theta|D) = f_p(\theta) \times f_L(D|\theta)$
- $\frac{\partial f(\theta)}{\partial \theta_i} = \frac{f(\theta_i+\delta)f(\theta_i-\delta)}{2\delta}$
- $q(x,y) = exp(-\frac{1}{4\tau}||x - y - \tau\nabla(log(f(y)))||^2_2)$
- $\epsilon \sim \mathcal{N}(0, 0.6)^d$
- $\delta = 0.01$
- $N = 250$
- $\tau = 0.0001$

Convergence of the Model





- Acceptance ratio : around 60% for each model.
- Coefficient Smaller than previous Methods. Smaller differences between teams.

Results

- Same procedure to obtain predictions.
- Home Advantage coefficient is high for both models : $\Delta \approx 0.40$.

- Poisson Model.
- 51.58% of well predicted results.
- Results distributed as follows :
  - H = 338
  - D = 34
  - A= 8

- Skellam Model.
- 52.05% of well predicted results.
- Results distributed as follows :
  - H = 292
  - D =71
  - A=17

- Too much weight on home coefficients. Optimize overall prediction but undermines team level.
  $\Rightarrow$ Need to augment number of iterations and use different approach calculate the gradient.

Conclusion Metropolis Adjusted-Langeving

- Advantage : Percentage of well predicted result is a bit higher.
- Drawbacks : Very costly methods : slow iteration.
  $\Rightarrow$Need to compute twice the gradient at each iteration.
- Need to find τ and $Var(\varepsilon)$ that gives good acceptance ratio.
- Model : For this method Skellam distribution gives better results.

Backtracking line search Gradient Descent

Algorithm :

Input $\theta_0, N_{max}, r, \alpha$

for i in 1 :N_{max} :

1. Calculate $\nabla(f(\theta_i))$
2. Update $\theta^c_{i+1} = \theta_i - r \times \nabla(f(\theta_i))$
3. While $f(\theta^c_{i+1}) >$ $f(\theta_{i+1}) - \times r \times ||\nabla(f(\theta_i))||^2$:
   - $r = \alpha \times r$
   - $\theta^c_{i+1} = \theta_i - r \times \nabla(f(\theta_i))$

Output : ${\theta^c}_{N_{Max}}$

Implemented :

- $\theta_0 \sim \mathcal{N}(0, 0.1)^{39}$
- $f(\theta) = -log(f_p(\theta) \times f_L(D|\theta))$
- $\frac{\partial f(\theta)}{\partial \theta_i} = \frac{f(\theta_i+\delta) - f(\theta_i-\delta)}{2\delta}$
- $\delta = 0.01$
- $r = 5x10^-3$
- $\alpha = 0.2$
- $N_{max} = 500$

Results

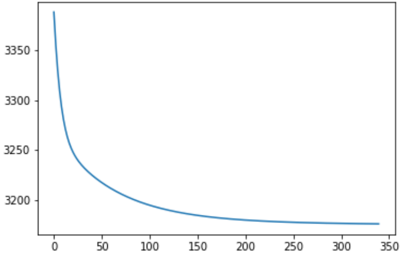


FIGURE – Minimization of the negative log likelihood

- Poisson Model.
- 50.79% of well predicted results.
- Home coefficient is : $\Delta = 0.39$
- Results distributed as follows :
  - H=248
  - D = 64
  - A= 68
- Similar to Metropolis Within Gibbs results.

Conclusion Backtracking line search Gradient Descent

- Advantage : Method easily implemented with satisfying results.
- Easily adapted to other models.
- Can be used to compute other methods as Stochastic Gradient Descent.
- Drawbacks : Computationally slow, approximate the gradient at each iteration.
- Need to find initial coefficient θ_0 and rate r that minimizes the negative log-likelihood.

Comparing Methods

- Analyze the coefficient given to each team with Poisson posterior.
- For each methods similar home Advantage coefficient.
- Calculate the probability of team A to beat team B without considering effects of home coefficient.
- Results are given as 20×20 matrix. Tables are found in Annexes.

Results

$\Rightarrow$ Similar results obtained for method 1 and method 3.

$\Rightarrow$ Distinction between teams is smaller with MALA method.

Gives worse results.

$\Rightarrow$ Results reflect well the reality for the best 4 and worse 4 teams.

$\Rightarrow$ Harder to obtain good results for the mid table teams.

Comparing Methods

Team	Atalanta	Bari	Bologna	Cagliari	Catania	Chievo	Fiorentina	Genoa	Inter	Juventus	Lazio	Livorno	Milan	Napoli	Palermo	Parma	Roma	Sampdoria	Siena	Udinese
Atalanta	0	0,323468	0,412146	0,344593	0,368024	0,357069	0,271713	0,33313	0,163102	0,237584	0,327003	0,477813	0,212732	0,313816	0,302274	0,412203	0,214793	0,320598	0,392243	0,345851
Bari	0,366332	0	0,425829	0,360625	0,381551	0,369056	0,286857	0,349713	0,17898	0,25548	0,34187	0,489169	0,228838	0,329404	0,320085	0,426416	0,232705	0,336334	0,40524	0,363618
Bologna	0,28285	0,26041	0	0,277507	0,300958	0,293115	0,215845	0,266892	0,123097	0,183549	0,263701	0,40116	0,16464	0,25146	0,23877	0,338942	0,16425	0,257127	0,32284	0,276323
Cagliari	0,357436	0,329979	0,419494	0	0,374876	0,363609	0,277547	0,339968	0,167412	0,243302	0,333534	0,485453	0,217835	0,320273	0,308887	0,419622	0,220177	0,327161	0,399296	0,353018
Catania	0,312682	0,288332	0,368712	0,307361	0	0,318399	0,242005	0,297019	0,145837	0,211614	0,291505	0,429218	0,189664	0,279676	0,26926	0,368774	0,191404	0,285766	0,350489	0,308389
Chievo	0,304057	0,280414	0,356172	0,299108	0,31763	0	0,236577	0,289614	0,146493	0,209552	0,2833	0,41279	0,187815	0,272529	0,264005	0,356576	0,19051	0,278402	0,33828	0,301135
Fiorentina	0,419874	0,389182	0,479062	0,414139	0,432295	0,416787	0	0,403495	0,220044	0,305496	0,392407	0,541488	0,273841	0,380331	0,374144	0,480492	0,281218	0,38793	0,456428	0,420183
Genoa	0,37328	0,344815	0,436571	0,367204	0,390742	0,378877	0,290666	0	0,176586	0,255817	0,34844	0,503501	0,228994	0,33491	0,323663	0,436814	0,231846	0,34205	0,415721	0,369186
Inter	0,590914	0,553849	0,649374	0,585075	0,598055	0,576597	0,490074	0,574867	0	0,466124	0,556932	0,70836	0,420874	0,545324	0,545728	0,652297	0,437401	0,554508	0,6232	0,597372
Juventus	0,494975	0,460399	0,559897	0,488593	0,508609	0,491463	0,397689	0,47675	0,26349	0	0,464028	0,626339	0,327267	0,450421	0,443835	0,561497	0,336411	0,459049	0,535201	0,495774
Lazio	0,360942	0,333482	0,419985	0,355283	0,376128	0,363824	0,282395	0,344455	0,175862	0,251242	0	0,482946	0,22505	0,324426	0,315079	0,420539	0,228753	0,331271	0,399616	0,358136
Livorno	0,223048	0,205045	0,274129	0,218306	0,241272	0,236363	0,167509	0,208677	0,089905	0,137641	0,208055	0	0,123901	0,196894	0,183604	0,272898	0,121702	0,201487	0,260639	0,215087
Milan	0,508917	0,47419	0,56937	0,502967	0,519025	0,500147	0,413425	0,492237	0,285491	0,386041	0,477435	0,631684	0	0,465249	0,462126	0,571614	0,358862	0,473844	0,544311	0,512346
Napoli	0,381135	0,352421	0,441809	0,375298	0,396418	0,3834	0,29916	0,364169	0,187645	0,267205	0,355826	0,506106	0,239326	0	0,333881	0,44249	0,243657	0,350261	0,420636	0,378687
Palermo	0,416197	0,385182	0,482065	0,40982	0,433387	0,419895	0,326791	0,397596	0,202739	0,290814	0,388952	0,55077	0,260283	0,374861	0	0,48265	0,264705	0,382638	0,459614	0,413171
Parma	0,287704	0,264861	0,345769	0,282232	0,306424	0,298607	0,219293	0,27134	0,124306	0,186018	0,268243	0,408704	0,166805	0,255664	0,242502	0	0,166272	0,26144	0,328801	0,280879
Roma	0,528073	0,492285	0,592459	0,521703	0,540447	0,522074	0,427982	0,510021	0,289696	0,396229	0,495862	0,657755	0,35599	0,482435	0,477318	0,594389	0	0,491363	0,567103	0,530228
Sampdoria	0,376262	0,347793	0,437364	0,370391	0,392083	0,37946	0,29462	0,359146	0,183155	0,262038	0,35124	0,502135	0,234672	0,338362	0,328607	0,437916	0,238531	0	0,416388	0,373344
Siena	0,289752	0,266952	0,344601	0,284569	0,306148	0,297296	0,222792	0,274391	0,131186	0,192446	0,270085	0,404162	0,172591	0,258415	0,247253	0,344357	0,173273	0,264138	0	0,28454
Udinese	0,373011	0,344288	0,43936	0,366646	0,392522	0,381452	0,288498	0,354199	0,170373	0,250751	0,348151	0,509267	0,224262	0,333744	0,3207	0,439249	0,226031	0,340987	0,418493	0

FIGURE – Table of probability for Metropolis within Gibbs Methods

- Probability of row team to beat column team.
- Darker the cell higher are the odds of winning.
- We see clear distinction between the strongest and weakest team.

Conclusion

- In this project better to choose Poisson Model : as efficient but computationally faster, simpler and easy interpretability.
- Metropolis Within Gibbs methods is computationally faster than other methods and easy to interpret.

To go further

- Change the prior distribution to incorporate prior knowledge on the teams.
- Augment number of iteration $\Rightarrow$ optimize coefficients.
- Could try to take into account the difference of the teams within the years (transfers) or the date when the match is played.
  $\Rightarrow$ More motivation towards end of season for teams near relegation or in Top 4 and tiredness from Champions League game.

References

Dimitris Karlis and Ioannis Ntzoufras (2008)

Bayesian modelling of football outcomes : Using the Skellam's distribution for the goal difference.

Department of Statistics, Athens University of Economics and Business

G. Dehaene (2019)

Bayesian Computation.

EPFL

Annexes

- For the Metropolis Within Gibbs method the table is given by : Link:Table Metropolis within Gibbs.
- For the Metropolis Adjusted-Langeving the table is given by : Link:Table MALA.
- For the Backtracking Gradient descent the table is given by : Link:Table Gradient Descent.