High Resolution Land Surface Parameter Estimation using Earth Observation technologies and Machine Learning

HeRLS PEUEOTAML


KEVIN HART
WHAT NOW?
NOW PLAYING IN THEATERS

So I should be talking about ....

- Drought monitoring in East Africa
- Land Surface Models
- Earth Observation Data
- Evapotranspiration, Precipitation and Soil Moisture

Another time …

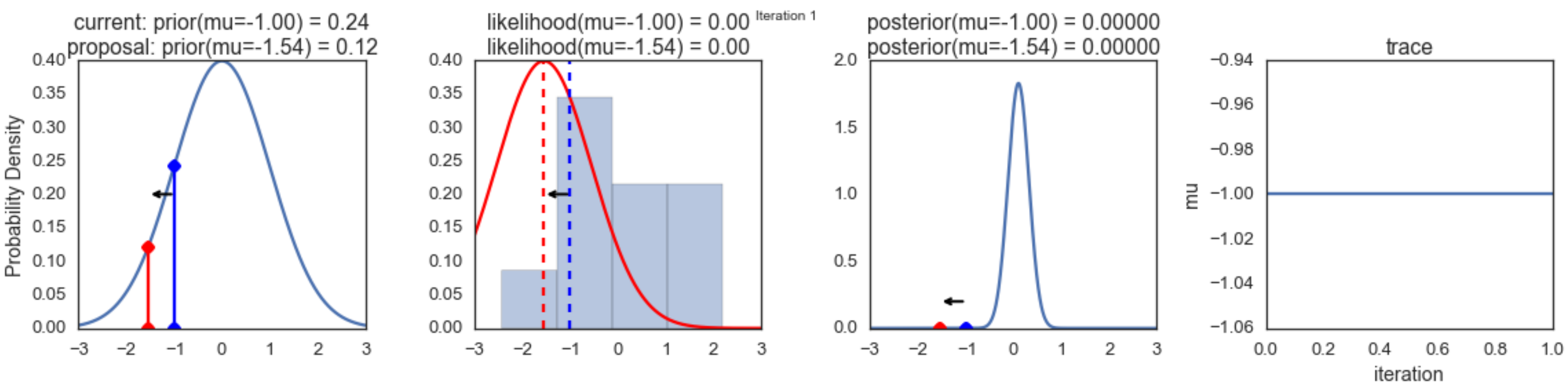
Modelling the outcome of football matches using Bayesian Statistics

Bayesian Statistics

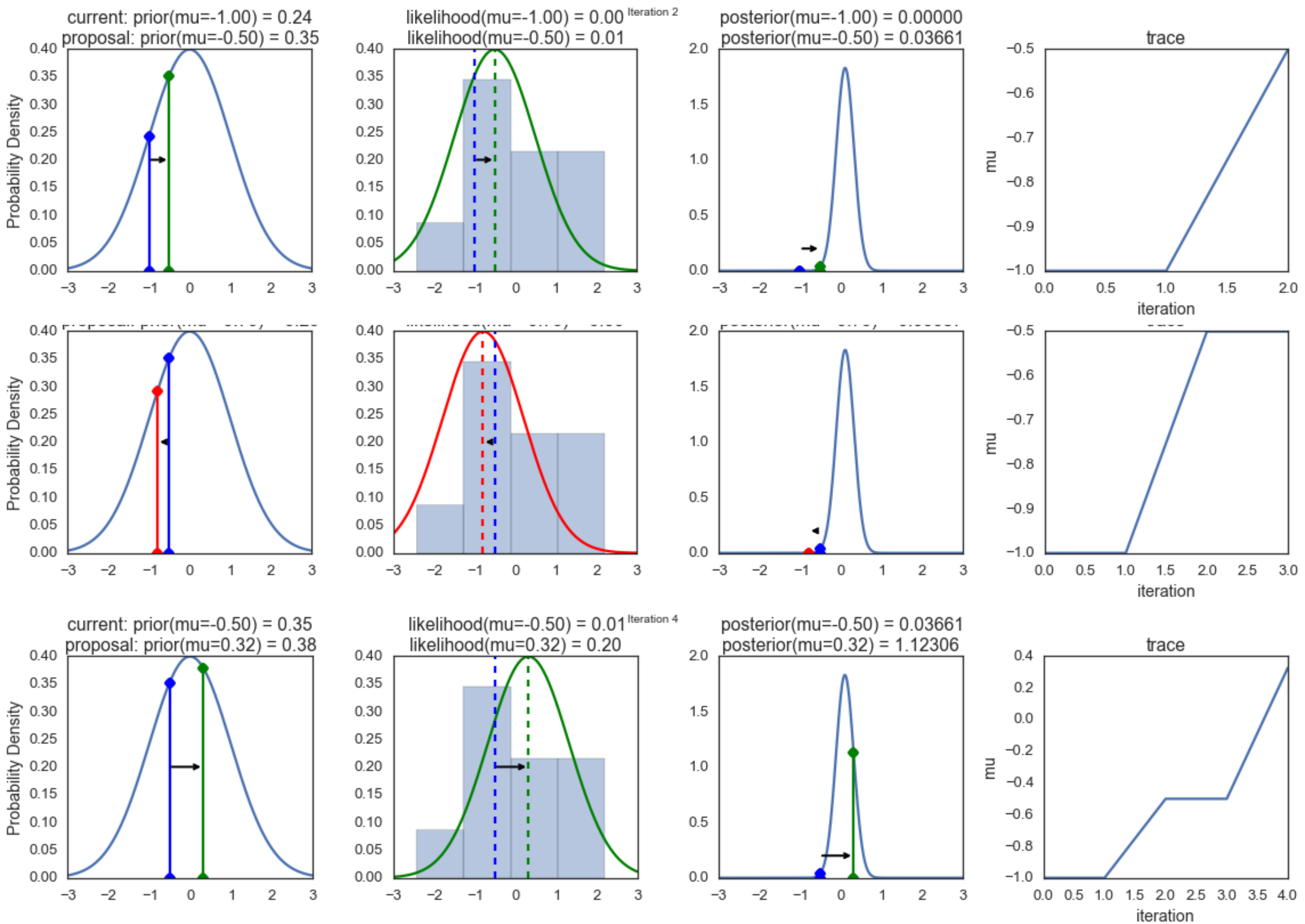
- Specify your prior distributions.
- Develop a generative model (likelihood)
  - A conditional probability distribution
  - P(Data | Parameters)
- Run an MCMC sampler.
- Return a posterior distribution.
- Check the model outputs.

Stan

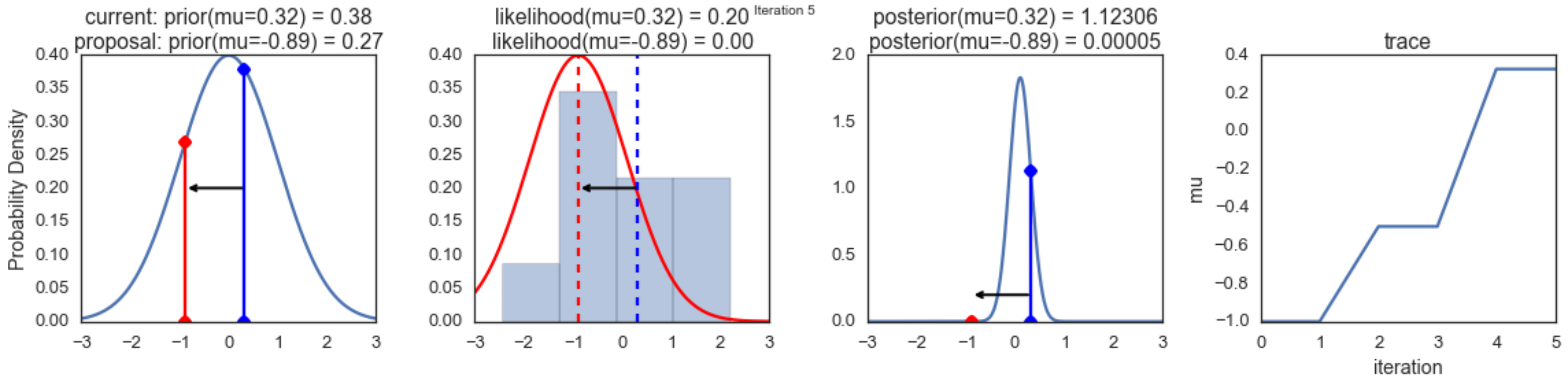
- Hamiltonian Monte Carlo Sampler
- Amazing online help
  - https://discourse.mc-stan.org/
- Defines a statistical model through a conditional probability function $p(\theta|y,x)$
- Probabilistic Programming



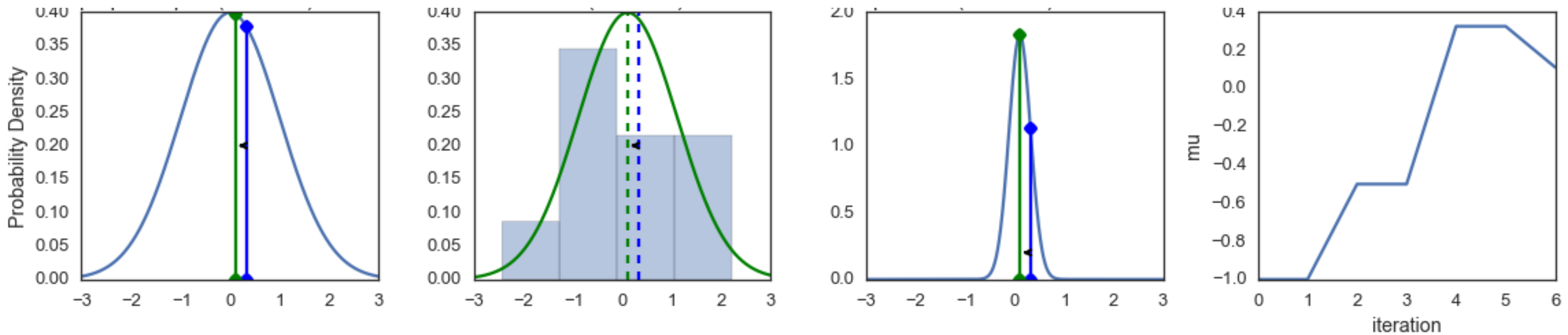
current: prior(mu=-1.00) = 0.24
proposal: prior(mu=-1.54) = 0.12
Probability Density
likelihood(mu=-1.00) = 0.00
likelihood(mu=-1.54) = 0.00
Iteration 1
posterior(mu=-1.00) = 0.00000
posterior(mu=-1.54) = 0.00000
trace
mu
iteration



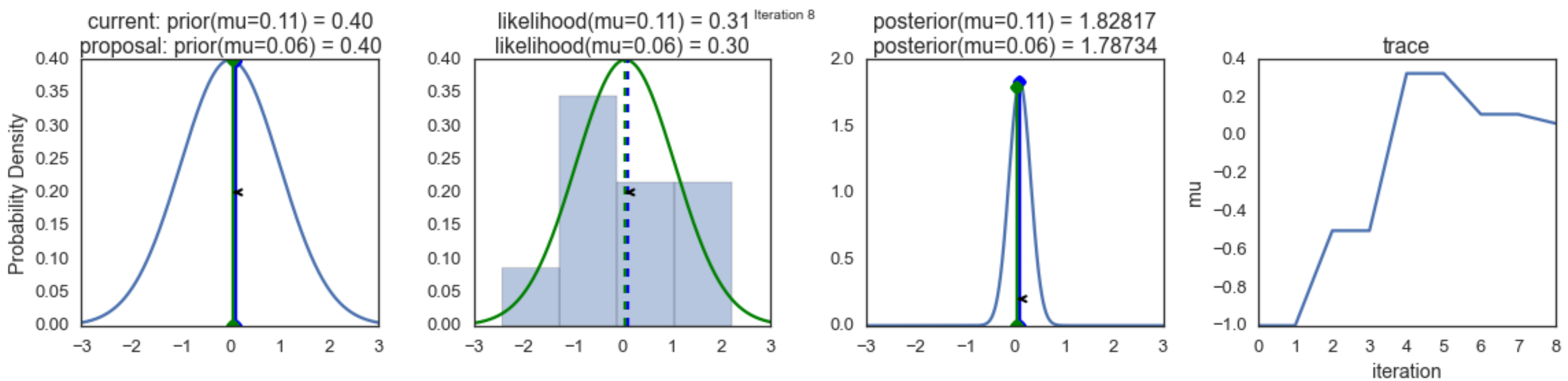
current: prior(mu=-1.00) = 0.24
proposal: prior(mu=-0.50) = 0.35
likelihood(mu=-1.00) = 0.00
likelihood(mu=-0.50) = 0.01
Iteration 2
posterior(mu=-1.00) = 0.00000
posterior(mu=-0.50) = 0.03661
trace
Probability Density
mu
iteration
current: prior(mu=-0.50) = 0.35
proposal: prior(mu=0.32) = 0.38
likelihood(mu=-0.50) = 0.01
likelihood(mu=0.32) = 0.20
Iteration 4
posterior(mu=-0.50) = 0.03661
posterior(mu=0.32) = 1.12306
trace



current: prior(mu=0.32) = 0.38
proposal: prior(mu=-0.89) = 0.27
Probability Density
likelihood(mu=0.32) = 0.20
likelihood(mu=-0.89) = 0.00
Iteration 5
posterior(mu=0.32) = 1.12306
posterior(mu=-0.89) = 0.00005
trace
mu
iteration



Probability Density
mu
iteration



current: prior(mu=0.11) = 0.40
proposal: prior(mu=0.06) = 0.40
Probability Density
likelihood(mu=0.11) = 0.31
likelihood(mu=0.06) = 0.30
Iteration 8
posterior(mu=0.11) = 1.82817
posterior(mu=0.06) = 1.78734
trace
mu
iteration

2 teams / match

380 games.

20 teams / league

3 promoted.

3 relegated

Premier League

Can we model future performance as a function of past performance?

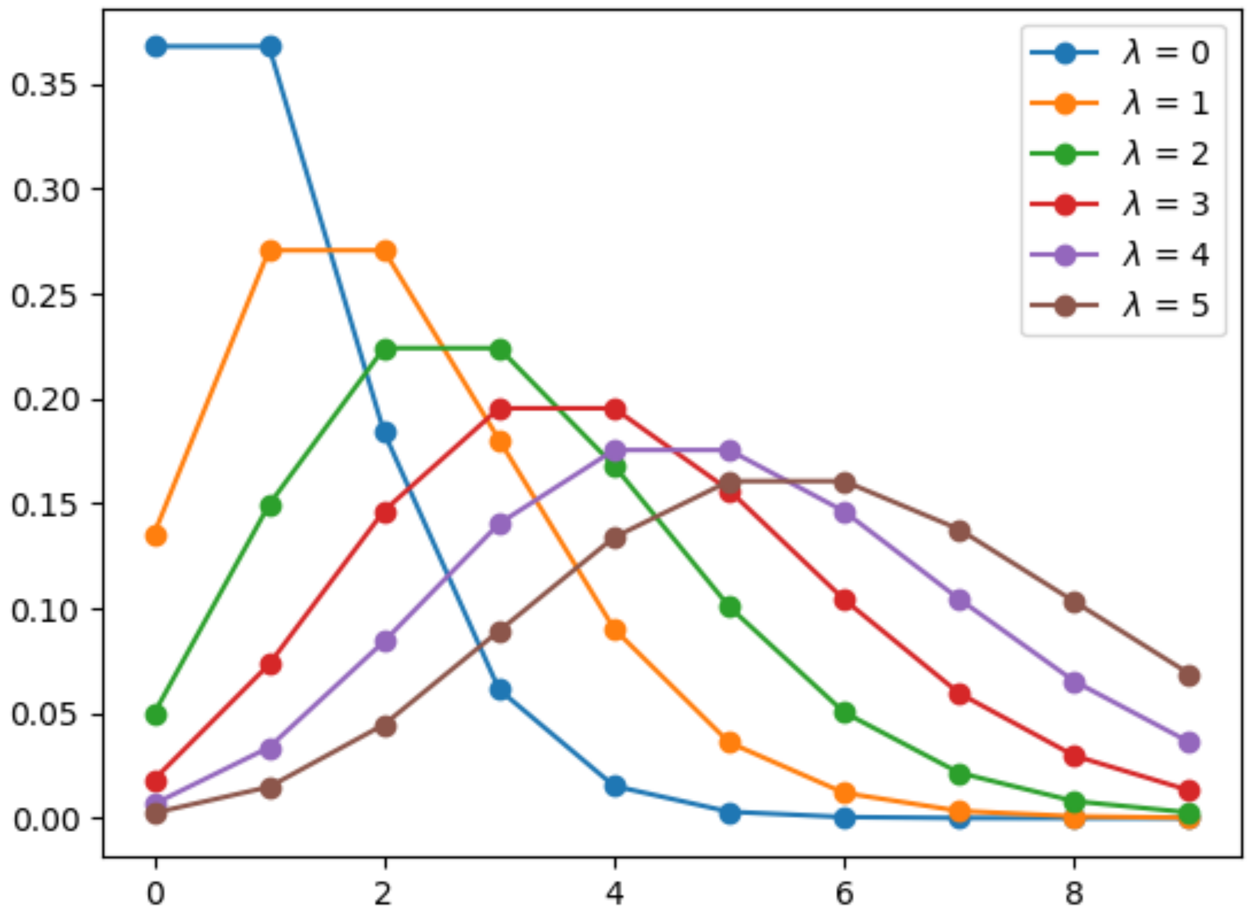
We can try.

- Two independent Poisson Distributions

- HG ~ Poisson(λ_ij)
- AG ~ Poisson(μ_ij)

- λ_ij = log(HA + Offense_i + Defence_j)
- μ_ij = log(Offense_j + Defence_i)

- Parameters:
  - 1 HA for the league
  - 2 parameters per team
- Constraints:
  - Attack and defence scores must sum to 0



0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
λ = 0
λ = 1
λ = 2
λ = 3
λ = 4
λ = 5

````
```stan
  // Priors (uninformative)
  offense ~ normal(0, 10);
  defense ~ normal(0, 10);
  home_advantage ~ normal(-10, 100);

  for (g in 1:n_games) {
    home_expected_goals[g] = exp(offense[home_team[g]] + defense[away_team[g]] + home_advantage);
    away_expected_goals[g] = exp(offense[away_team[g]] + defense[home_team[g]]);

    home_goals[g] ~ poisson(home_expected_goals[g]);
    away_goals[g] ~ poisson(away_expected_goals[g]);
  }
```
````

Live Demo …

What have we captured?

1. Unique ‘skill’ scores for each team
2. Skill as a product of attack and defense
3. Outcomes the result of two teams relative to one another
4. Estimate home advantage (but don’t assume it exists or is even positive)

So let's celebrate

What have we missed?

1. Scores are not independent
2. Lower scoring games are under-predicted
3. There is no time varying element in the model

What a load of rubbish …

Loads of interesting work …

- https://twiecki.github.io/blog/
- http://opisthokonta.net/
- http://pena.lt/y/
- https://web.archive.org/web/20150526184248/http://www.sportshacker.net/posts/
- https://betanalpha.github.io/assets/case_studies/principled_bayesian_workflow.html

- Maher (1982)
- Dixon and Coles (1994)
- Karlis and Ntzoufras (2012)

$$y_n = \alpha + \beta x_n + \epsilon_n \quad \text{where} \quad \epsilon_n \sim \mathsf{Normal}(0, \sigma).$$

This is equivalent to the following sampling involving the residual,

$$y_n - (\alpha + \beta X_n) \sim \mathsf{Normal}(0, \sigma),$$

and reducing still further, to

$$y_n \sim \mathsf{Normal}(\alpha + \beta X_n, \sigma).$$

This latter form of the model is coded in Stan as follows.

```
data {
  int<lower=0> N;
  vector[N] x;
  vector[N] y;
}
parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}
model {
  y ~ normal(alpha + beta * x, sigma);
}
```