THE BETTING MACHINE

Bli søkkrik på fotballtipping med kunstig intelligens

Universitetet i Oslo / Dagen@IFI
Thomas Andresen
31/10/13

Thomas Andresen

Sivilingeniør i datateknikk, NTNU
Konsulent i BEKK (juni 2013)

AGENDA

STATISTISKE DATA

BAYESISKE MODELLER

MONTE CARLO SIMULATION

MARKOVKJEDER

MARKOV CHAIN MONTE CARLO (MCMC)

EKSEMPELMODELL

BETTING

FASIT PREMIER LEAGUE 13/14

STATISTISKE DATA

Hvilke data snakker vi om?

- Kampoppsett
- Resultater fra tidligere kamper (antall mål)
- Mannskap (skader, spilleforbud osv.)
- Kampstatistikk
  - Skudd på mål, redningsprosent, ballbesittelse, duellseire osv.
- Spillestil (kontrings- vs maskinfotball)
- +++

Hva bruker vi dataene til?

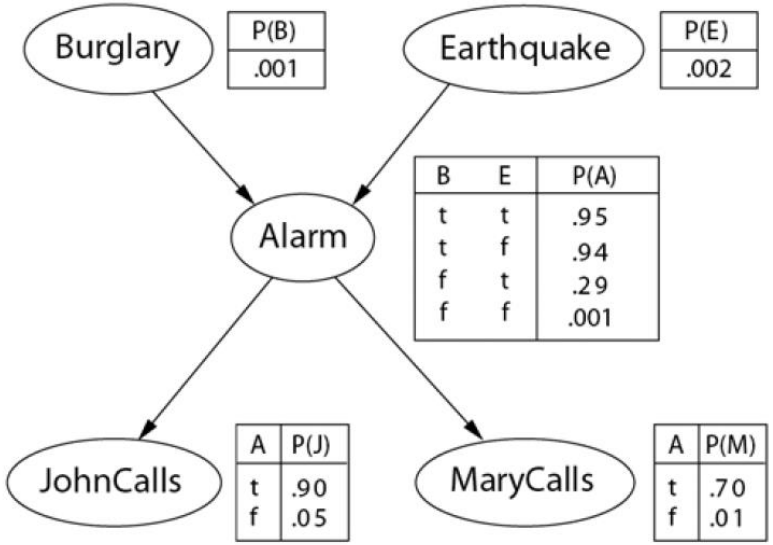
- Lager matematiske modeller som bruker fortiden til å forsøke å fortelle noe om fremtiden
  - Hvor mange mål scorer et lag i en gitt kamp?
  - Hvem vinner PL 13/14?
- Maskinlæring (supervised learning)

Hvor henter vi dataene?

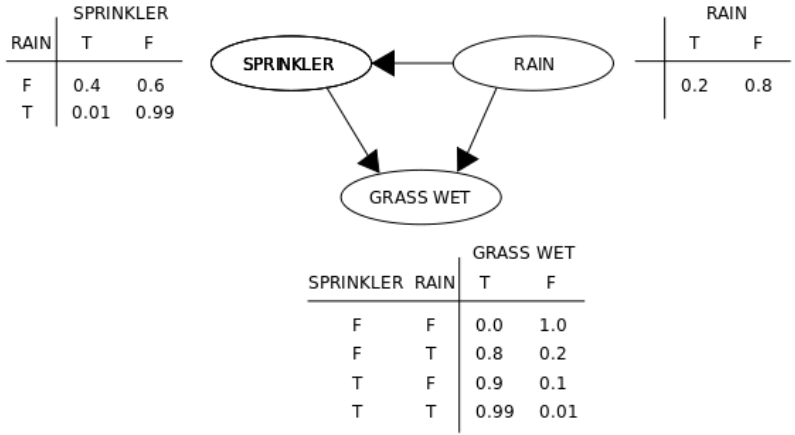
- Mange tjenester ($$$)
  - OptaPro
  - WhoScored
  - OddsPortal
- Fortsatt populært med tippesystemer – marked for å selge data.
- Noen få tjenester som tilbyr data gratis og tilgjengelig (API e.l)
  - football-data.co.uk (CSV)

Hvordan modellere kunnskapen vi har om fortiden?

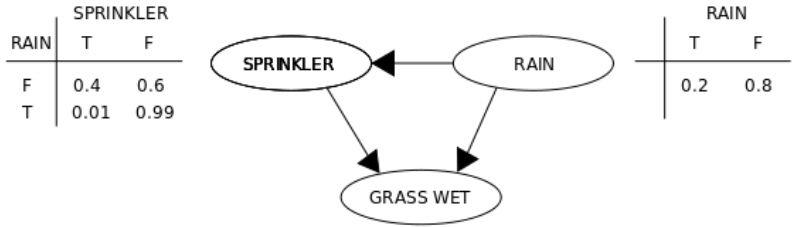
- Bayesiske modeller
  - Probabilistiske – kan reflektere ulike grader av sikkerhet/usikkerhet.
  - Enkelt å modellere ulike sammenhenger i dataene.
  - Visuelle (graf)



- Består av noder og kanter med retning (directed edges)
- Nodene representerer variabler i modellen og kantene sier noe om forholdet mellom de (parent -> child).
- DAG – Directed Acyclic Graph
- Variablene (random/stochastic variable) består av et sett med ulike verdier og deres tilhørende sannsynlighet.
  - Verdiene kan også være en kontinuerlig sannsynlighetsfordeling (sannsynlighetstetthet).
  - Betingede sannsynligheter
    - P(Grass Wet|Rain).



- Forholdsvis enkelt å gjøre inferens ved hjelp av Bayes teorem
  - Kan karakteriseres som en mekanisme for å benytte teoremet på komplekse problemer.



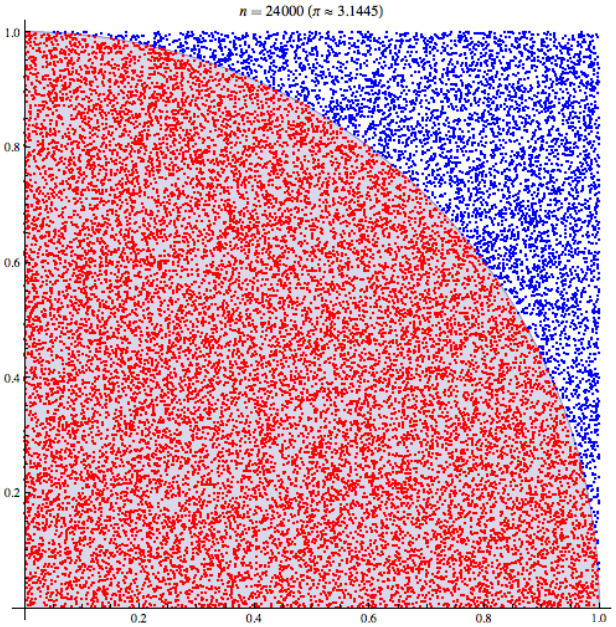
RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99

RAIN T	RAIN F
0.2	0.8

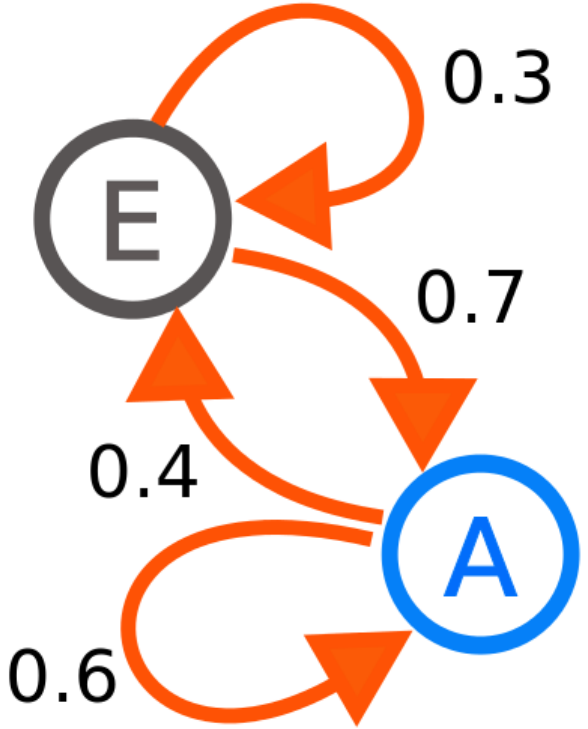
SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

MONTE CARLO SIMULATION

- Teknikk utviklet i forbindelse med Manhattanprosjektet på 1940-tallet.
- Kort fortalt: Tilfeldig (uniform) sampling
  - Derav koblingen til Monte Carlo (terninger)
- Mange bruksområder
  - Løse integraler i høyere dimensjoner
  - Sampling
  - Optimalisering
- Løser deterministiske problemer på en probabilistisk måte

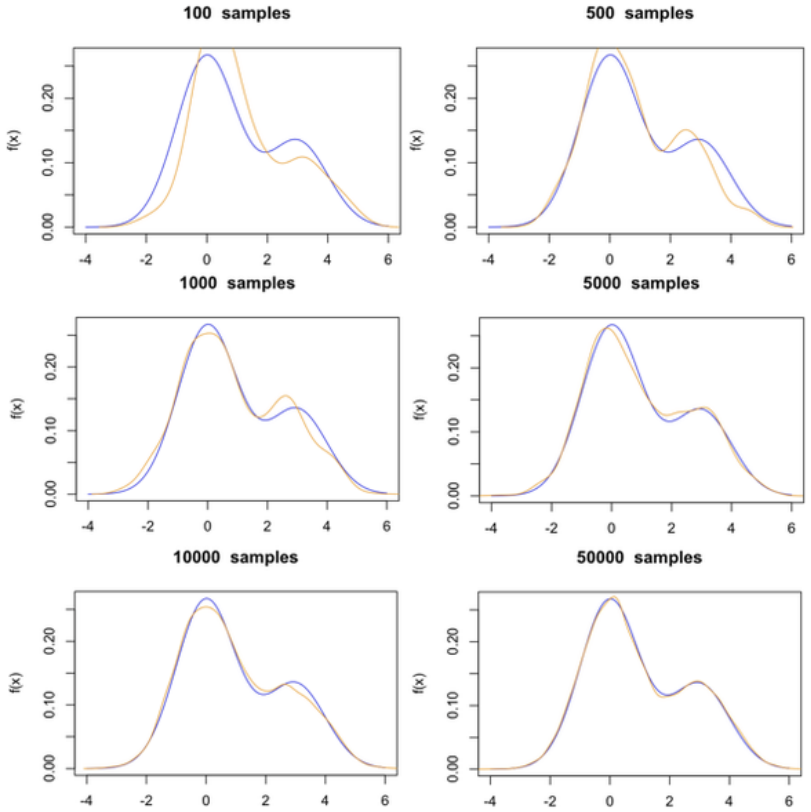


- Matematisk modell for stokastiske systemer med tilstander som har tilhørende sannsynligheter (transition probability).
- Nåværende tilstand av markovkjeden er utelukkende avhengig av tidligere tilstander.
  - I en markovkjede av 1. orden er nåværende tilstand kun avhengig av den forrige.
- Er lett å konstruere og har mange ønskede egenskaper dersom kjeden er ergodisk (er asyklisk og alle noder kan nåes fra hvor som helst i systemet).
  - Konvergerer til en stasjonær sannsynlighetsfordeling.



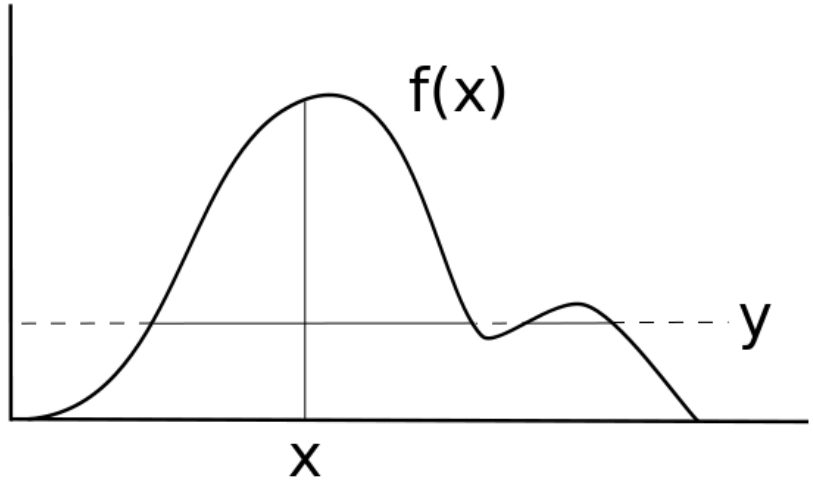
MARKOV CHAIN MONTE CARLO (MCMC)

- En klasse med algoritmer som indirekte sampler fra sannsynlighetsfordelinger ved å konstruere en markovkjede som konvergerer til den ønskede fordelingen.
- Som regel en siste utvei dersom problemet er for komplekst til å kunne løses med andre metoder.
  - Tilnærmingsmetode som krever mange samples for å oppnå et brukbart resultat.



MARKOV CHAIN MONTE CARLO (MCMC)

- Mange teknikker under samme paraply
  - Gibbs sampling
  - Slice sampling
  - Rejection sampling
  - Metropolis-Hastings
- Må gjerne kombinere flere teknikker for å løse komplekse problemer.



MARKOV CHAIN MONTE CARLO (MCMC)

- Jeg sa tidligere at det var enkelt å gjøre inferens på Bayesiske modeller – hvorfor forteller jeg nå om sampling og terninger?
- For å utføre Bayesisk inferens, må vi ha den posteriore sannsynlighetsfordelingen til modellen.
  - Samlet betinget sannsynlighet etter observasjoner er gjort (evidence).
- … og slik regner vi oss frem til den:

$$P(\theta, D) = \frac{P(\theta)P(D|\theta)}{P(D)} = \frac{P(\theta)P(D|\theta)}{\int P(\theta)P(D|\theta)d\theta} \propto P(\theta)P(D|\theta)$$

MARKOV CHAIN MONTE CARLO (MCMC)

- Krevende å regne ut for selv trivielle modeller – umulig for de mer komplekse.

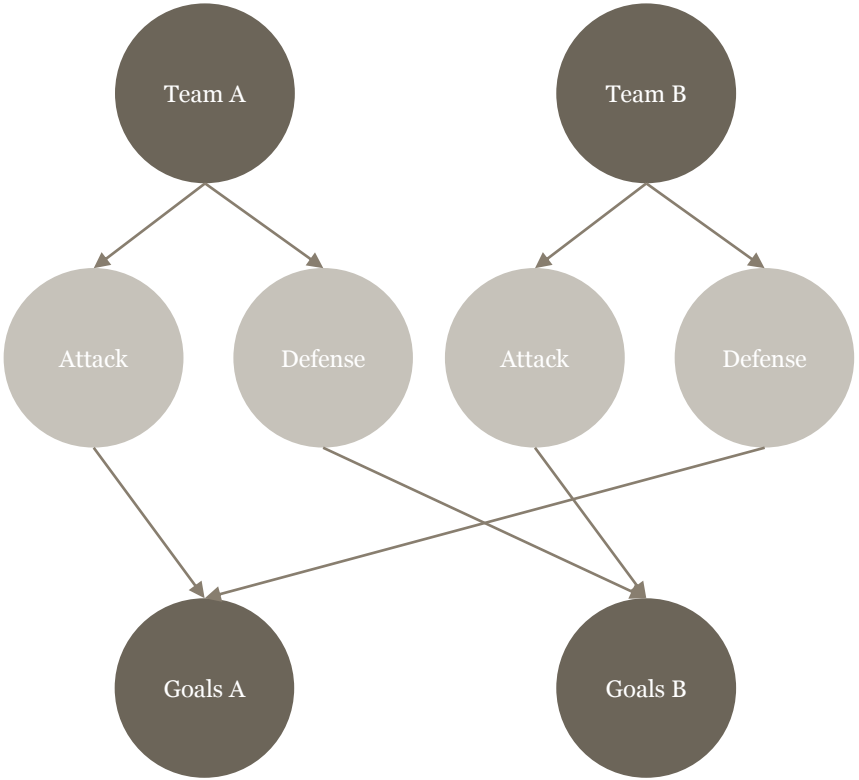
EKSEMPELMODELL

Skal se på en veldig enkel modell jeg har laget for anledningen.

- Bruker minimalt med data – tar kun hensyn til antall mål scoret i tidligere kamper.
- Rask kjøretid og gir overraskende gode resultater.

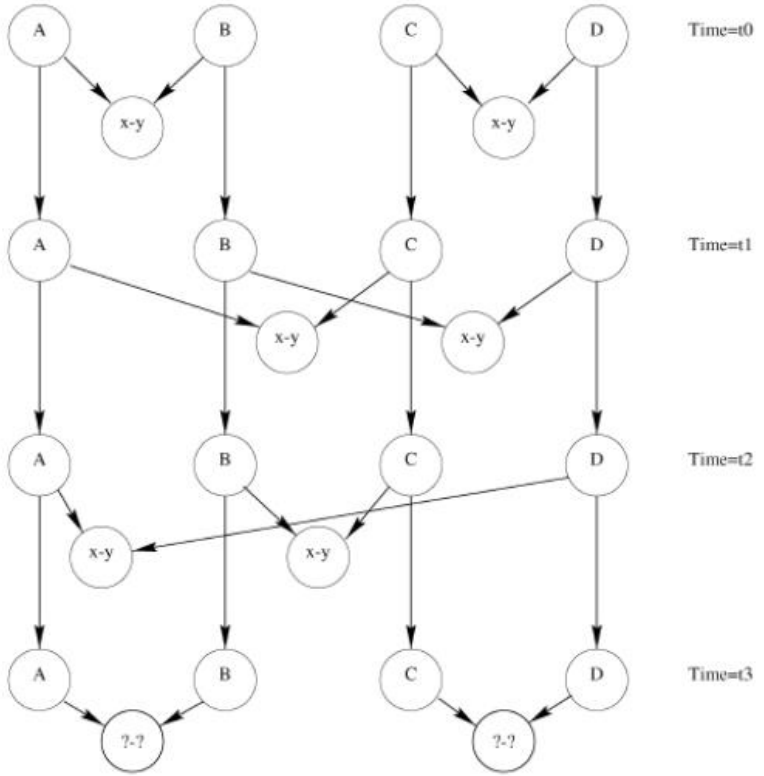
EKSEMPELMODELL - MÅLMODELL

- Antall mål scoret av et lag i en kamp er Poissonfordelt
  - Diskret sannsynlighetsfordeling
  - God fit (Maher, 1982)
- Antall mål scoret av et lag i en kamp er avhengig av angrepsferdighetene til laget selv og forsvarsferdighetene til motstanderen.
- Ferdighetene (attack og defense) utvikler seg over tid (wiener process). Dette er skjulte (hidden) nodes.



EKSEMPELMODELL – FULL MODELL

- I en liga (PL) spiller lagene mot hverandre 2 ganger i løpet av en sesong – hjemme og borte. Dette kalles gjerne Round-robin scheduling.
- Kan være litt vrient - Alle kampene må tilpasses i et hierarki etter hvilken hvilken rekkefølge de spilles i.
  - Må ha flere konsepter for rekkefølge og kunne mappe mellom de forskjellige:
    - Kampnummer (1-380)
    - Runde/slice-nummer (1-38)



EKSEMPELMODELL

- Vi skal utvikle en modell som kan kjøres av programmet JAGS (Just Another Gibbs Sampler).
  - Videreutvikling av BUGS
  - Cross-platform
  - Open source
  - Modeller utrykkes i et eget språk
  - Data leses og skrives i R-format
- mcmc-jags.sourceforge.net

EKSEMPELMODELL - IMPLEMENTASJON

```
data {
        homeGoalAvg <- 0.395
        awayGoalAvg <- 0.098
}

model {
        precision ~ dgamma(10, 0.2)

        for(t in 1:noTeams) {
                attack[t, 1] ~ dnorm(0, precision)
                defense[t, 1] ~ dnorm(0, precision)

                for(s in 2:noTimeslices) {
                        attack[t, s] ~ dnorm(attack[t, (s-1)], precision)
                        defense[t, s] ~ dnorm(defense[t, (s-1)], precision)
                }
}
```

EKSEMPELMODELL - IMPLEMENTASJON

```
gamma ~ dunif(0, 0.1)

for(i in 1:noGames) {
        delta[i]<- (attack[team[i, 1], timeslice[i, 1]] + defense[team[i, 1], timeslice[i, 1]] –
        attack[team[i, 2], timeslice[i, 2]] - defense[team[i, 2], timeslice[i, 2]]) / 2

        log(homeLambda[i]) <- (homeGoalAvg + (
                attack[team[i, 1], timeslice[i, 1]] -
                defense[team[i, 2], timeslice[i, 2]] -
                gamma * delta[i]))

        log(awayLambda[i]) <- (awayGoalAvg + (
                attack[team[i, 2], timeslice[i, 2]] –
                defense[team[i, 1], timeslice[i, 1]] +
                gamma * delta[i]))

        goalsScored[i, 1] ~ dpois(homeLambda[i])
        goalsScored[i, 2] ~ dpois(awayLambda[i])
```

Hvor kommer maskinlæringsbiten inn?

- Inferens og læring i JAGS er tett sammenkoblet (i Bayesiske modeller generelt)
  - I første del av kjøring (training data)
    - Modellen lærer seg fordelingene av de skjulte variablene (attack, defense) fra de komplette dataene (hvor Goals A og Goals B er definert for de respektive kampene)
  - I andre del av kjøring (test/validation data)
    - Goals A og Goals B settes til NaN (JAGS tolker dette som en hittil ukjent verdi)
    - Antall mål infereres fra hvert lags respektive attack og defense variabler

EKSEMPELMODELL - RESULTAT

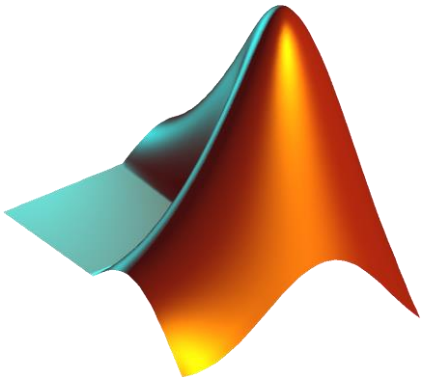
Jeg får endelig modellen til å kompilere og kjøre. Hva gjør jeg med resultatet?

- JAGS benytter et system av monitorer som fanger opp samples fra parameterne de er tilordnet.
  - Dumpes til filer på disken.
  - Samplene representerer den tilnærmet betingede posteriore sansynlighetsfordelingen til parameteren som ble monitorert
    - Aggregere og tegne et histogram
    - Beregne forventningsverdi (sum), standardavvik og varianse
    - Sammenligne mot samples fra andre parametre (Goals A mot Goals B) – en estimator på sannsynligheten for at lag A vinner/taper.
      - Teller antall ganger en sample fra A er større enn en sample fra B og deler på det totale antallet samples => sannsynlighet for hjemmeseier.

JAGS - WRAPPERE

Det finnes wrappere til JAGS for flere ulike platformer.

- Forenkler behandlingen av samples
- Abstraherer bort CLI
- Parallelisering



- rjags
  JAGS-wrapper for R
- matjags
  JAGS-wrapper for Matlab
- SharpJags
  JAGS-wrapper for .NET
  github.com/thrandre/SharpJags
  (evt. Nuget)

Hva er formålet med disse modellene?

- Tippe riktig resultat i alle kamper og bli millionær over natten.

Eller

- Finne de kampene som er feilpriset i markedet (odds) og på sikt tjene en slant.

Hva er formålet med disse modellene?

- ~~Tippe riktig resultat i alle kamper og bli millionær over natten.~~

Eller

- Finne de kampene som er feilpriset i markedet (odds) og på sikt tjene en slant.

Feilpriset i markedet? Hæ?

- Bookmakere setter prisen på sine spilleobjekter basert på matematiske modeller og «ekspert»-kunnskap.
  - Opening odds
  - Closing odds – styrt av kjøp og salg i markedet
- Tjene 10% = På sikt tippe resultatet 10% mer nøyaktig enn bookmakeren.
- Spilleobjekter med closing odds omsettes i et effektivt marked (Efficient-market hypothesis)
- Vil plassere bets så tidlig som mulig – utnytte ineffektiviteter.

Simulering av de 10 siste kampene i PL

HomeTeam	AwayTeam	P(H)	P(D)	P(A)	O(H)	O(D)	O(A)	B_Max_Outcome	B_Max	ExpectedReturn	Variance	Stake	R	Profit
Chelsea	Man City	0.47	0.21	0.32	2.40	3.00	3.20	H	0.063	0.130	1.435	0.697	H	1.672
Sunderland	Newcastle	0.28	0.19	0.53	2.90	3.00	2.62	A	0.137	0.400	1.708	0.585	H	-0.585
Swansea	West Ham	0.52	0.23	0.25	1.75	3.40	5.00	A	0.011	0.231	4.638	0.216	D	-0.216
Tottenham	Hull	0.46	0.29	0.26	1.44	4.00	8.00	D	0.014	0.145	3.269	0.306	H	-0.306
Aston Villa	Everton	0.37	0.24	0.39	3.40	3.00	2.30	H	0.034	0.245	2.683	0.373	A	-0.373
Crystal Palace	Arsenal	0.20	0.17	0.63	9.00	4.50	1.36	H	0.005	0.793	12.924	0.077	A	-0.077
Liverpool	West Brom	0.47	0.26	0.27	1.44	4.50	6.50	D	0.012	0.190	3.938	0.254	H	-0.254
Man United	Stoke	0.46	0.29	0.26	1.36	4.33	10.00	D	0.016	0.238	3.828	0.261	H	-0.261
Norwich	Cardiff	0.44	0.24	0.33	2.25	3.10	3.40	A	0.017	0.110	2.542	0.393	D	-0.393
Southampton	Fulham	0.50	0.27	0.23	1.65	3.40	6.00	A	0.010	0.404	6.454	0.155	H	-0.155

Tabellen ved sesongslutt

Team	Points
Arsenal	94
Chelsea	92
Man City	88
Southampton	81
Liverpool	74
Everton	72
Swansea	68
West Brom	61
Tottenham	61
Fulham	55
West Ham	54
Hull	53
Newcastle	53
Man United	53
Aston Villa	46
Stoke	41
Cardiff	39
Norwich	23
Sunderland	7
Crystal Palace	6

Kildekode (JAGS og C#) er tilgjengelig på

github.com/thrandre/TheBettingMachine

SPØRSMÅL?

TAKK FOR MEG!