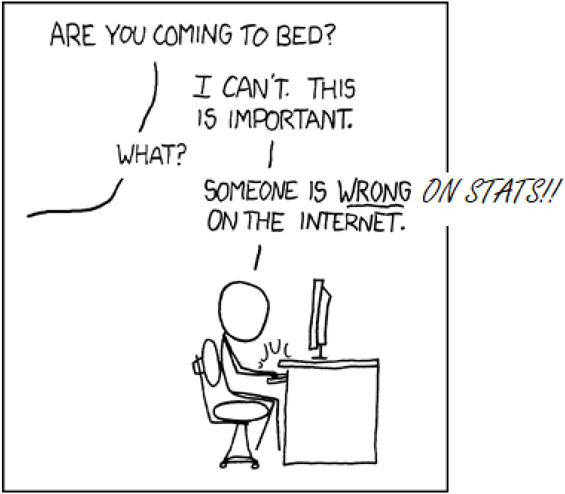
Using TMB (Template Model Builder) to predict the winner of a ping pong match

Alex Fun
@quicklingbard
github.com/alexfun

20180712 (3.50 pm @ useR!2018/P7)

Part 1: The stats.stackexchange problem



(adapted from www.xkcd.com/386)

Cross Validated

QUESTIONS TAGS USERS BADGES UNANSWERED ASK QUESTION

If I have a 58% chance of winning a point, what's the chance of me winning a ping pong game to 21, win by 2?

85

34

I have a bet with a co-worker that out of 50 ping pong games (first to win 21 points, win by 2), I will win all 50. So far we've played 15 games and on average I win 58% of the points, plus I've won all the games so far. So we're wondering if I have a 58% chance of winning a point and he has a 42% chance of winning a point, what's the percent chance that I would win the game? Is there a formula that we can plug in difference % chances?

We've googled all over and even asked the data scientists at our company but couldn't find a straight answer.

Edit: Wow, I am blown away by the thoroughness of responses. Thank you all so much!!! In case people are curious, I have an update to how my bet is going: I've now won 18 out of 50 games, so I need to win 32 more games. I've won 58.7% of all points and my opponent has therefore won 41.3% of points. The standard deviation for my opponent is 3.52, his average score is 14.83, and his median score is 15.50. Below is a screenshot of the score of each game so far. I can keep updating as the bet goes on, if people are interested.

Edit #2: Unfortunately we've only been able to play a few more games, below are the results. I'm just going to keep replacing the picture so I don't have a bunch of screenshots of the score.

Final Update: I finally lost to my co-worker on game #28. He beat me 21-13. Thanks for all of your help!

Game #	My Opponent	Me	Winner
1	18	21	Me
2	17	21	Me
3	11	21	Me
4	13	21	Me
5	15	21	Me
6	15	21	Me
7	16	21	Me

asked 4 months ago

viewed 13,670 times

active 3 months ago

Linked

7 Competing negative binomials

Related

10 If a tennis match was a single large set, how many games would give the same accuracy?

11 How can I predict the odds that a dodgeball team is going to win based on the winning history of its players?

4 Statistics of 7 game playoff series

1 World Cup: Is it worth to bet each match twice?

1 Optimal strategy in growing jackpot scenario

6 Chance of me beating my friend in trivia

2 How do I model the probability of a win in a particular matchup, given historical win/loss data for the players

4 How to properly develop a machine learning model for a poker game?

1 Efficiently computing poisson binomial sum

3 Why don't contestants use this strategy when playing Secret X on the Price is Right

(https://stats.stackexchange.com/questions/329521)

Cross Validated

QUESTIONS | TAGS | USERS | BADGES | UNANSWERED | ASK QUESTION

Qn:

If I have a 58% chance of winning a point, what's the chance of me winning a ping pong game to 21, win by 2?

85

34

I have a bet with a co-worker that out of 50 ping pong games (first to win 21 points, win by 2), I will win all 50. So far we've played 15 games and on average I win 58% of the points, plus I've won all the games so far. So we're wondering if I have a 58% chance of winning a point and he has a 42% chance of winning a point, what's the percent chance that I would win the game? Is there a formula that we can plug in difference % chances?

We've googled all over and even asked the data scientists at our company but couldn't find a straight answer.

Edit: Wow, I am blown away by the thoroughness of responses. Thank you all so much!!! In case people are curious, I have an update to how my bet is going: I've now won 18 out of 50 games, so I need to win 32 more games. I've won 58.7% of all points and my opponent has therefore won 41.3% of points. The standard deviation for my opponent is 3.52, his average score is 14.83, and his median score is 15.50. Below is a screenshot of the score of each game so far. I can keep updating as the bet goes on, if people are interested.

Edit #2: Unfortunately we've only been able to play a few more games, below are the results. I'm just going to keep replacing the picture so I don't have a bunch of screenshots of the score.

Final Update: I finally lost to my co-worker on game #28. He beat me 21-13. Thanks for all of your help!

Game #	My Opponent	Me	Winner
1	18	21	Me
2	17	21	Me
3	11	21	Me
4	13	21	Me
5	15	21	Me
6	15	21	Me
7	16	21	Me
8	9	21	Me

asked 4 months ago

viewed 13,670 times

active 3 months ago

Linked

7 Competing negative binomials

Related

10 If a tennis match was a single large set, how many games would give the same accuracy?

11 How can I predict the odds that a dodgeball team is going to win based on the winning history of its players?

4 Statistics of 7 game playoff series

1 World Cup: Is it worth to bet each match twice?

1 Optimal strategy in growing jackpot scenario

6 Chance of me beating my friend in trivia

2 How do I model the probability of a win in a particular matchup, given historical win/loss data for the players

4 How to properly develop a machine learning model for a poker game?

1 Efficiently computing poisson binomial sum

3 Why don't contestants use this strategy when playing Secret X on the Price is Right

Qn:

If I have a 58% chance of winning a point, what's the chance of me winning a ping pong game to 21, win by 2?

85

34

I have a bet with a co-worker that out of 50 ping pong games (first to win 21 points, win by 2), I will win all 50. So far we've played 15 games and on average I win 58% of the points, plus I've won all the games so far. So we're wondering if I have a 58% chance of winning a point and he has a 42% chance of winning a point, what's the percent chance that I would win the game? Is there a formula that we can plug in difference % chances?

We've googled all over and even asked the data scientists at our company but couldn't find a straight answer.

Edit: Wow, I am blown away by the thoroughness of responses. Thank you all so much!!! In case people are curious, I have an update to how my bet is going: I've now won 18 out of 50 games, so I need to win 32 more games. I've won 58.7% of all points and my opponent has therefore won 41.3% of points. The standard deviation for my opponent is 3.52, his average score is 14.83, and his median score is 15.50. Below is a screenshot of the score of each game so far. I can keep updating as the bet goes on, if people are interested.

Edit #2: Unfortunately we've only been able to play a few more games, below are the results. I'm just going to keep replacing the picture so I don't have a bunch of screenshots of the score.

Final Update: I finally lost to my co-worker on game #28. He beat me 21-13. Thanks for all of your help!

Data:

Game #	My Opponent	Me	Winner
1	18	21	Me
2	17	21	Me
3	11	21	Me
4	13	21	Me
5	15	21	Me
6	15	21	Me
7	16	21	Me

asked 4 months ago

viewed 13,670 times

active 3 months ago

Linked

- 7 Competing negative binomials

Related

- 10 If a tennis match was a single large set, how many games would give the same accuracy?
- 11 How can I predict the odds that a dodgeball team is going to win based on the winning history of its players?
- 4 Statistics of 7 game playoff series
- 1 World Cup: Is it worth to bet each match twice?
- 1 Optimal strategy in growing jackpot scenario
- 6 Chance of me beating my friend in trivia
- 2 How do I model the probability of a win in a particular matchup, given historical win/loss data for the players
- 4 How to properly develop a machine learning model for a poker game?
- 1 Efficiently computing poisson binomial sum
- 3 Why don't contestants use this strategy when playing Secret X on the Price is Right

Cross Validated

QUESTIONS TAGS USERS BADGES UNANSWERED ASK QUESTION

If I have a 58% chance of winning a point, what's the chance of me winning a ping pong game to 21, win by 2?

Assume:

85

34

I have a bet with a co-worker that out of 50 ping pong games (first to win 21 points, win by 2), I will win all 50. So far we've played 15 games and on average I win 58% of the points, plus I've won all the games so far. So we're wondering if I have a 58% chance of winning a point and he has a 42% chance of winning a point, what's the percent chance that I would win the game? Is there a formula that we can plug in difference % chances?

We've googled all over and even asked the data scientists at our company but couldn't find a straight answer.

Edit: Wow, I am blown away by the thoroughness of responses. Thank you all so much!!! In case people are curious, I have an update to how my bet is going: I've now won 18 out of 50 games, so I need to win 32 more games. I've won 58.7% of all points and my opponent has therefore won 41.3% of points. The standard deviation for my opponent is 3.52, his average score is 14.83, and his median score is 15.50. Below is a screenshot of the score of each game so far. I can keep updating as the bet goes on, if people are interested.

Edit #2: Unfortunately we've only been able to play a few more games, below are the results. I'm just going to keep replacing the picture so I don't have a bunch of screenshots of the score.

Final Update: I finally lost to my co-worker on game #28. He beat me 21-13. Thanks for all of your help!

Game #	My Opponent	Me	Winner
1	18	21	Me
2	17	21	Me
3	11	21	Me
4	13	21	Me
5	15	21	Me
6	15	21	Me
7	16	21	Me

asked 4 months ago

viewed 13,670 times

active 3 months ago

Linked

7 Competing negative binomials

Related

10 If a tennis match was a single large set, how many games would give the same accuracy?

11 How can I predict the odds that a dodgeball team is going to win based on the winning history of its players?

4 Statistics of 7 game playoff series

1 World Cup: Is it worth to bet each match twice?

1 Optimal strategy in growing jackpot scenario

6 Chance of me beating my friend in trivia

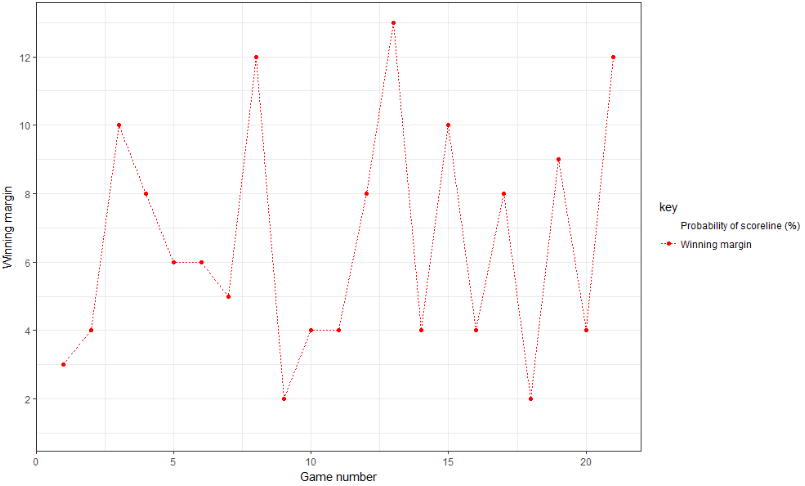
2 How do I model the probability of a win in a particular matchup, given historical win/loss data for the players

4 How to properly develop a machine learning model for a poker game?

1 Efficiently computing poisson binomial sum

3 Why don't contestants use this strategy when playing Secret X on the Price is Right

But what if the assumption is wrong?



But what if the assumption is wrong?

- Postulate that in a truly competitive game, the original poster has a much higher (than 59%) chance of winning a point

Part 2: Why use automatic derivatives in optimisation?

Optimisation

- Objective function (e.g. log-likelihood) and decision variables (e.g. regression coefficients).
- Gradient based vs non-gradient based (not relevant here).

Optimisation: Newton's method

Suppose one wishes to find the value of (the decision variables) $\mathbf{x}$ such that $y = f(\mathbf{x})$ is maximised. Start with an initial guess $\mathbf{x}_0$ for the optimal values, and iteratively build better guesses $\mathbf{x}_1, \mathbf{x}_2, \ldots$, via:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \frac{f'(\mathbf{x}_n)}{f''(\mathbf{x}_n)}$$

Optimisation: Newton's method

Suppose one wishes to find the value of (the decision variables) $\mathbf{x}$ such that $y = f(\mathbf{x})$ is maximised. Start with an initial guess $\mathbf{x}_0$ for the optimal values, and iteratively build better guesses $\mathbf{x}_1, \mathbf{x}_2, \ldots$, via:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \frac{f'(\mathbf{x}_n)}{f''(\mathbf{x}_n)}$$

If $\mathbf{x}$ is a vector:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - [\mathbf{H}f(\mathbf{x}_n)]^{-1}\nabla f(\mathbf{x}_n)$$

(https://en.wikipedia.org/wiki/Newton%27s_method_in_optimization)

Optimisation: Newton's method

Nabla is the gradient and **H** is the hessian:

$$\nabla = \left(\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_m} \right)$$

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_m} \\ \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_m\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_m\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_m^2} \end{bmatrix}$$

Optimisation: Newton's method

Nabla is the gradient and **H** is the hessian:

$$\nabla = \left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_m}\right)$$

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_m} \\ \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_m\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_m\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_m^2} \end{bmatrix}$$

Wow, much derivatives to find! (LaTeX from Wikipedia)

Calculating derivatives: Analytically

- Logistic regression:

$$\log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots$$

Calculating derivatives: Analytically

- Logistic regression:

$$\log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots$$

- Log-likelihood:

$$\ell = \sum_{i=1}^{n} y_i \log(\frac{p_i}{1-p_i}) + \sum_{i=1}^{n} \log(1-p_i)$$

Calculating derivatives: Analytically

- Logistic regression:

$$\log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots$$

- Log-likelihood:

$$\ell = \sum_{i=1}^{n} y_i \log(\frac{p_i}{1-p_i}) + \sum_{i=1}^{n} \log(1-p_i)$$

- After some algebra (or consultation of https://stats.stackexchange.com/a/16541):

$$\begin{aligned}\frac{\partial \ell}{\partial \beta_j} &= ... \\ &= \sum_i y_i x_{ij} - \sum_i p_i x_{ij},\end{aligned}$$

$$\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} = \dots$$

Calculating derivatives: Analytically

- Logistic regression:

$$\log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots$$

- Log-likelihood:

$$\ell = \sum_{i=1}^{n} y_i \log(\frac{p_i}{1-p_i}) + \sum_{i=1}^{n} \log(1-p_i)$$

- After some algebra (or consultation of https://stats.stackexchange.com/a/16541):

$$\begin{aligned}\frac{\partial \ell}{\partial \beta_j} &= ... \\ &= \sum_i y_i x_{ij} - \sum_i p_i x_{ij},\end{aligned}$$

$$\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k} = \dots$$

(ellipses are left as an exercise for the reader.)

How can one calculate derivatives?

- By hand (aka analytic derivatives)

How can one calculate derivatives?

- By hand (aka analytic derivatives)
- Symbolic differentiation (Mathematica, Sage, other software packages)

How can one calculate derivatives?

- By hand (aka analytic derivatives)
- Symbolic differentiation (Mathematica, Sage, other software packages)
- Numerical differentiation (computational by picking a small value for Δx):

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

How can one calculate derivatives?

- By hand (aka analytic derivatives)
- Symbolic differentiation (Mathematica, Sage, other software packages)
- Numerical differentiation (computational by picking a small value for Δx):

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

- Automatic differentiation

Why automatic differentiation?

- It's magic! (since minimal additional code or thought is required to get derivatives)

Why automatic differentiation?

- It's magic! (since minimal additional code or thought is required to get derivatives)
- Exact to machine precision.

Why automatic differentiation?

- It's magic! (since minimal additional code or thought is required to get derivatives)
- Exact to machine precision.
- Efficient: $\text{cost}(\nabla f) \leq 4\,\text{cost}(f)$.

Why automatic differentiation?

- It's magic! (since minimal additional code or thought is required to get derivatives)
- Exact to machine precision.
- Efficient: $\text{cost}(\nabla f) \leq 4\,\text{cost}(f)$.
- https://justindomke.wordpress.com/2009/02/17/automatic-differentiation-the-most-criminally-underused-tool-in-the-pe

Part 3: TMB

What is TMB

- Website for installation and tutorial:
  https://github.com/kaskr/adcomp/wiki
- Support:
  https://groups.google.com/forum/#!forum/tmb-users

What is TMB

- Website for installation and tutorial:
  https://github.com/kaskr/adcomp/wiki
- Support:
  https://groups.google.com/forum/#!forum/tmb-users
- R package that calculates first and second derivatives of any objective function written in C++.
- Very, very fast $\leftrightarrow$ uses state of the art C++ packages like Eigen, CppAD, ...
- Used where more ‘complex’ log-likelihood functions are required. E.g.:
  - the glmmTMB package allows you to “fit linear and generalized linear mixed models with various extensions, including zero-inflation”.
  - A Laplace approximation for the likelihood is implemented to deal with models containing random effects.

Modelling: Probability to win a point

- Let s_i be the observed score difference in the i-th game.

Modelling: Probability to win a point

- Let s_i be the observed score difference in the i-th game.
- Postulate that the probability of the original poster winning the $i+1$-th game is:

$$p_{i+1} = \alpha p_0 + (1-\alpha)(\mathrm{expit}(\mathrm{logit}(p_i) + \beta))$$

Modelling: Probability to win a point

- Let s_i be the observed score difference in the i-th game.
- Postulate that the probability of the original poster winning the $i+1$-th game is:

$$p_{i+1} = \alpha p_0 + (1-\alpha)(\text{expit}(\text{logit}(p_i) + \beta))$$

where α is a 'mixing' coefficient:

$$\alpha = \text{expit}(\alpha_0 + \alpha_1 s_i)$$

Modelling: Probability to win a point

- Let s_i be the observed score difference in the i-th game.
- Postulate that the probability of the original poster winning the $i+1$-th game is:

$$p_{i+1} = \alpha p_0 + (1-\alpha)(\text{expit}(\text{logit}(p_i) + \beta))$$

  where α is a 'mixing' coefficient:

$$\alpha = \text{expit}(\alpha_0 + \alpha_1 s_i)$$

- The aim is to find values for the following parameters that maximise the likelihood of observing the game scorelines:
  - p_0: probability of point win when original poster is not playing at their best.
  - β: changes probability of winning in the next game depending on the scoreline this game.
  - α_0 and α_1: coefficients governing the averaging between p_0 and p_i.

Modelling: Probability to win a game with a certain scoreline

- In a game, suppose player one wins points with probability p, and let x and y be the total number of points player one and two won, respectively.
- Suppose player one reaches 21 points first with a margin of two:

$$\Pr(21 : y) = \binom{20 + y}{y} p^{21} (1 - p)^{y}$$

Modelling: Probability to win a game with a certain scoreline

- In a game, suppose player one wins points with probability p, and let x and y be the total number of points player one and two won, respectively.
- Suppose player one reaches 21 points first with a margin of two:

$$\Pr(21 : y) = \binom{20 + y}{y} p^{21}(1 - p)^{y}$$

- Else:

$$\Pr(x : y) = 2^{y-20}\binom{40}{20} p^{x}(1 - p)^{y}$$

(whuber: https://stats.stackexchange.com/a/329554)

Code: R

```r
R_obj_fun_template <- function(par, me, you, score_diff) {

    p_0 <- par[1]
    alpha_0 <- par[2]
    alpha_1 <- par[3]
    beta <- par[4]

    n_obs <- length(score_diff)

    p <- p_0 # initialise initial model fitted probability
    nll <- 0 # initialise model negloglik

    for (i in seq_along(you)) {

        if (i > 1) {
            # mixing coefficient
            alpha <- plogis(alpha_0 + alpha_1 * score_diff[i - 1]);

            # update p with model
            p <-  alpha * p_0 + (1 - alpha) *  plogis(qlogis(p) + beta);


        }

        # increment nll
        nll <- nll - log(get_prob_of_scoreline(me[i], you[i], p));

    }



    return(nll)
}

optim_in_R$fn <- function(par) R_obj_fun_template(par, df$Me, df$Opponent, df$score_diff)
optim_in_R$par <- c(0.592246, 0, 0, 0)
# quasi-Newton optimisation with approximations to the Hessian
# i.e. second derivatives are never required
optim_in_R$method <- 'BFGS'
```

Code: R

```r
get_prob_of_scoreline <- function(me, you, prob_win_point){

    # warning, not vectorised!
    stopifnot(all(length(me) == 1, length(you) == 1, length(prob_win_point) == 1))

    min_me_you <- min(me, you)


    if (you >= 20) {
        out_prob <- choose(40, 20) * 2^(min_me_you - 20) * prob_win_point^me * (1 - prob_win_point)^you

    } else {
        out_prob <- choose(20 + min_me_you, min_me_you) * prob_win_point^me * (1 - prob_win_point)^you

    }

    return(out_prob);
}
```

Code: Rcpp

```
R_obj_fun_template_Rcpp <- cppFunction(
    "double R_obj_fun_templatep_Rcpp(NumericVector par, IntegerVector me, IntegerVector you, IntegerVector score_diff) {



    // parameters:
    double p_0 = par[0];
    double alpha_0 = par[1];
    double alpha_1 = par[2];
    double beta = par[3];

    int n_obs = me.size();

    double p = p_0; // initialise initial model fitted probability
    double nll = 0; // initialise model negloglik


    for (int i = 0; i < n_obs; i++) {

        if (i > 0) {
            // mixing coefficient
            double alpha = Rf_plogis(alpha_0 + alpha_1 * score_diff[i - 1], 0.0, 1.0, 1, 0);

            // update p with model

            p =  alpha * p_0 + (1 - alpha) *  Rf_plogis(Rf_qlogis(p, 0.0, 1.0, 1, 0) + beta, 0.0, 1.0, 1, 0);


        }
        nll += -log(get_prob_of_scoreline_Rcpp(me[i], you[i], p));


    }


    return nll;

}",
    includes = "double get_prob_of_scoreline_Rcpp(int me, int you, double prob_win_point) {
    double out_prob;
```

Code: TMB

```
template<class Type>
Type objective_function<Type>::operator() () {


    DATA_VECTOR(you);
    DATA_VECTOR(me);
    DATA_VECTOR(score_diff);


    // parameters:
    PARAMETER(p_0);
    PARAMETER(alpha_0);
    PARAMETER(alpha_1);
    PARAMETER(beta);

    int n_obs = me.size();

    Type p = p_0; // initialise initial model fitted probability
    Type nll = 0; // initialise model negloglik
    vector<Type> p_vec_out(n_obs);

    for (int i = 0; i < n_obs; i++) {

        if (i > 0) {
            // mixing coefficient
            Type alpha = invlogit(alpha_0 + alpha_1 * score_diff[i - 1]);

            // update p with model
            p =  alpha * p_0 + (1 - alpha) *  invlogit(logit(p) + beta);


        }
        nll += -log(get_prob_of_scoreline(me[i], you[i], p));
        // store into vector to be returned to R
        p_vec_out[i] = p;
    }


    REPORT(p_vec_out);
    return nll;

}
```

Example usage

```r
## Get TMB objective function which includes .$gr (gradient) and .$he (hessian) elements.
optim_in <- MakeADFun(data = list(me = df$Me, you = df$Opponent, score_diff = df$Me - df$Opponent),
                      parameters = list(p_0 = 0.592246, alpha_0 = 0, alpha_1 = 0, beta = 0),
                      DLL = 'reg_20180312_init_obj_fun',
                      silent = T)

# see that optim_in has the right elements to be passed into R's optimisation function optim.
str(optim_in)
# List of 10
#     $ par     : Named num [1:4] 0.592246 0 0 0
#     ..- attr(*, "names")= chr [1:4] "p_0" "alpha_0" "alpha_1" "beta"
#     $ fn      :function (x = last.par, ...)
#     $ gr      :function (x = last.par, ...)
#     $ he      :function (x = last.par, atomic = usingAtomics())
#     $ hessian : logi FALSE
#     $ method  : chr "BFGS"
#     $ retape  :function (set.defaults = TRUE)
#     $ env     :<environment: 0x000000001bd5b158>
#     $ report  :function (par = last.par)
#     $ simulate:function (par = last.par, complete = FALSE)


# optimise using optim
```

Results: Speed of optimisation

```
> microbenchmark(TMB = do.call(optim, optim_in),
+                TMB_numerical = do.call(optim, optim_in_no_jac),
+                Rcpp = do.call(optim, optim_in_Rcpp),
+                base_R = do.call(optim, optim_in_R))
Unit: milliseconds
          expr        min         lq       mean     median         uq
           TMB   4.028838   4.326319   4.737954   4.430616   4.755356
 TMB_numerical  16.559796  17.426758  18.959108  17.864092  18.819943
          Rcpp  15.682760  16.367006  18.124086  16.978561  18.956832
        base_R 553.979831 571.790193 607.814579 581.838190 609.109290
```

Results: Coefficients

```
> (optim_out <- do.call(optim, optim_in))
$`par`
        p_0     alpha_0     alpha_1        beta
  0.5461695 -33.9765827   3.7675724   0.1189786

$value
[1] 57.25596

$counts
function gradient
     161       68
```

Results: Coefficients

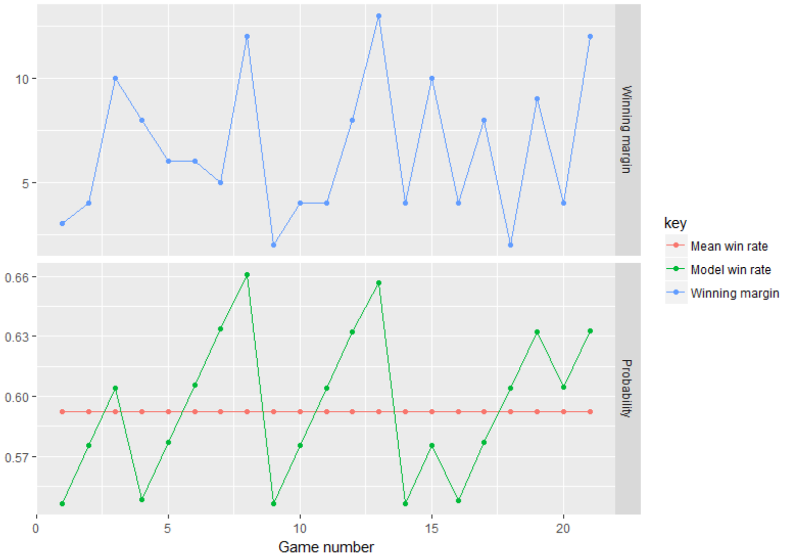
- Recall that s_i is the observed score difference in the i-th game.
- The probability of the original poster winning the $i+1$-th game, given by the model, is:

$$p_{i+1} = 0.546\alpha + (1-\alpha)(\mathrm{expit}(\mathrm{logit}(p_i) + 0.119))$$

where the mixing coefficient α is:

$$\alpha = \mathrm{expit}(-34 + 3.77 s_i)$$

Results: fitted game win probs



The last word

The last word

Your thoughts on whether points and games are independent? Not to mention stationary over time (cumulated games and experience playing each other)? – Mark L. Stone Mar 21 at 15:58

1 I don't think the points and games were independent. The more I won, the more complacent I got and my opponent took advantage of that. – richard Mar 21 at 19:28

Thank you

- TMB developers: Kasper Kristensen [aut, cre, cph], Hans Skaug [ctb], Arni Magnusson [ctb], Casper Berg [ctb], Anders Nielsen [ctb], Martin Maechler [ctb], Theo Michelot [ctb], Mollie Brooks [ctb], Alex Forrence [ctb], Christoffer Moesgaard Albertsen [ctb], Cole Monnahan [ctb]Thank you to user organisers + TMB developers
- useR!2018 organisers: Di Cook (Monash, chair), Rob J Hyndman (Monash), Nick Tierney (Monash), Earo Wang (Monash), Thomas Lumley (Auckland), Paul Murrell (Auckland), Bill Venables (Brisbane), Simone Blomberg (UQ), Kerrie Mengersen (QUT), Goknur Giner (WEHI), Jessie Roberts (QUT), Miles McBain (QUT), Eun-kyung Lee (EWHA, Korea), Paula Andrea (UQ), Wenjing Wang (RUC, China), Soroor Hediyehzadeh (WEHI), Rhydwyn McGuire (NSW DOH), Alex Whan (CSIRO), Steph de Silva (Rex Analytics), Anna Quaglieri (WEHI), Marie Trussart (WEHI), Kim Fitter (NZ), Monash Business School Events Team (Kerry, Hannah, Jess and Deb)