3504
Girls of Steel

Optimizing OPR with Weighted Least Squares

Gabriel Krotkov
Mentor

Anuva Ghosalkar
Strategy & Scouting Lead

Sienna Li
Data Science Lead

Anushka Prabhu
Data Science Team

Lily Tang
Data Science Team

Audrey Zheng
Data Science Team

Aashi Bhatt
Data Science Team

ABSTRACT

In this paper, we present an improvement to Offensive Power Rating (OPR), a popular linear regression model for assessing team performance at a given event. One key assumption of linear regression is the independence of the errors, but in the FIRST Robotics Competition (FRC) context, this assumption is not exactly true. Using data from all district events between 2009 and 2024, we model the unweighted errors as a function of tournament progression and generate weightings to improve the regression fit through Weighted Least Squares. This method consistently enhances OPR, advancing statistical insights within the FRC community.

1 Introduction

Every FRC qualification match consists of six randomly selected robots divided into two opposing alliances of three teams each. To strategize effectively and play to each robot's strengths, teams frequently need to make convenient, flexible, and accurate estimations of their partners' overall performance without relying on large amounts of scouted, robot-level data. OPR uses linear regression to estimate how many points an individual robot contributes to an alliance in any given match [1].

1.1 Background

Since Karthik Kanagasabapathy and Ian Mackenzie of FRC team 1114 developed "calculated contribution" in 2004, teams have been using linear regression to measure a team's scoring ability. Calculated contribution, generally considered the first application of linear regression in FRC, provides a less random method of analysis than average score by accounting for alliance partners. The first public description [1] of similar linear algebra came in 2006 from Scott Weingart of FRC team 293 on a Chief Delphi post which coined the term "Offensive Power Rating". Weingart's terminology for linear regression in FRC rose to popularity, commonly abbreviated as OPR. However, the usual formulation for the regression design matrix uses teams on the columns and matches on the rows (as described by Karthik [2]), where Weingart's formulation used teams on both the rows and the columns. In 2017, Eugene Fang detailed the math behind OPR in a blog post for TheBlueAlliance [3] ("TBA"), which has become the standard definition of linear regression for FRC.

1.2 OPR as Linear Regression

By definition, OPR is a multiple linear regression. To see this, reference Section 5.2 (pg 130-133) of Sheather [4], which shows the calculation behind multiple linear regression. Looking at the solved equation for β, $\hat{\beta} = (X'X)^{-1}X'Y$, we see that the TBA blog solved equation to find OPR, $x = M^{-1}s$, follows a very similar structure. With β and x as the OPR value, M and $(X'X)$ as the vector of alliance lineups, and s and $X'Y$ as the vector of alliance scores, Sheather's and TBA's explanations are identical. See [5] for a demonstration of the equivalence on data.

1.3 Motivation

Other metrics have so far proven more effective than OPR for match prediction [6], in large part because they incorporate historical data. For example, knowing that FRC team 254 has won 5 world championships is *very* relevant to making a good prediction about their performance in a given match. However, regression does not consider data from previous seasons or previous events. OPR will not see a difference between a multiple world champion and playoff bubble team unless it is given match data from the relevant event. This highlights the primary value of regression methods in FRC: summarizing team performance at a given event.[2] To improve the quality of our estimate, we can focus on the model's assumptions and their validity in an FRC context.

One of the key assumptions of linear regression is that the errors ε_i are independent and identically distributed with constant variance. That is, the prediction error for each match does not depend on any other matches, and that the spread of the errors does not change over the course of a tournament. In FRC, this is mostly true: matches are well isolated from each other, and the challenges that teams face over the course of the event do not change on average. However, teams do gain experience over the course of the event, make adjustments, and change strategies. Anecdotally, teams "settle in" to their most representative performance after their first few matches, with earlier matches contributing less to

[1] We will refer to a match as a qualification match competed by a particular alliance.

[2] This isn't to say we should not pay attention to match prediction! Match prediction is extremely useful as an empirical test for our methods; but if the goal is accurate match prediction, regression is not the best tool available.

overall performance. This dynamic likely influences the distribution of the errors so that it is not constant. Accounting for this non-constant error could improve the linear approximation of team quality.

1.4 Weighted Least Squares

Weighted Least Squares (WLS) is a statistical method used to improve regressions by modeling nonconstant residual variance. This method requires the user to know the variance structure of the response in order to model nonconstant variance - or to have a good approximation of it. Typically for cases that cannot be resolved by a transformation, each row of the design matrix is weighted proportional to the inverse variance of the errors (pg. 96, 97 [7]), giving less weight to the less precise observations. This theoretical quantity is usually best approximated by the inverse variance of the residuals[3].

WLS is generally implemented by dividing the diagonal of the regression covariance matrix by the weights, which directly downweights the least precise observations. This has computational advantages because it deals directly with the regression formulation. However, WLS can be equivalently implemented [8] by "row-replication", which duplicates rows an integer number of times to represent the additional weight placed on that row. This equivalent formulation highlights the intuition behind weighted least squares: putting additional importance on each row of the design matrix proportional to the size of the weight. Row-replication also has a flexibility advantage. Applying the weights to the covariance matrix requires a regression setting, while row-replication can be applied without the context of regression.

2 Methods

2.1 Data

Our analysis used qualification match data (alliance lineups and final total scores) from every district event that occurred between 2009 and 2024. Districts have exactly twelve qualification matches for each team, which makes comparison between events more consistent. To ensure that scores are comparable across different years, we standardized the scores of each event.

2.2 Weight Optimization

WLS allows us to remove the assumption of residual independence, if we can appropriately weight the rows of the design matrix. This means that we need a principled way to find weights that describe the distribution of the residuals. We computed descriptive weights in two ways: residual variance binning and linear weight smoothing. While each way provided a different set of weights, the second (linear weight smoothing), extends the first (residual variance binning).

2.2.1 Residual Variance Binning

Optimal weights for WLS are proportional to the inverse variance (see [7] pg. 97) of the error for that data point. To approximate this, we calculate the variance of the residuals of the unweighted linear model in six sequential bins. To bin the residuals, we give each match a "match percentile", which is $\frac{match\_number}{n\_matches}$; intuitively, this is the percentage of progress through the tournament at which the match takes place. Then, we evenly divide the alliance-matches into six bins based only on their match percentile. Since each team plays twelve matches, six bins allows for an average of two matches from each team to be in each corresponding bin. This balances the granularity to avoid hyper-analyzing the differences or failing to recognize the trend.

Taking the variance of the residuals in each bin provides a numerical way to measure the reliability of OPR. To reflect the general trend of the residuals with a unique set of weights, we take the reciprocal of each binned variance.

2.2.2 Linear Weight Smoothing

Variance binning directly approximates the model's variance; but this could under-smooth and overfit the data. To mitigate this, we also compute linearly smoothed weights, taking the linear regression of the residual variances. Linear weight smoothing significantly reduces the number of weight combinations to test on the OPR model when trying to find the optimal set of weights, providing a more efficient process than residual variance binning in most cases.

Linear weight smoothing continues from where variance binning leaves off. By graphing the residual variances against event progression percentile, we can then fit an appropriate function onto the coordinate points. With the residual

[3] A residual is defined as the difference between the value predicted by a linear regression model and the observed (true) value. The residual in our context is the difference between the score predicted by OPR and the actual alliance score.

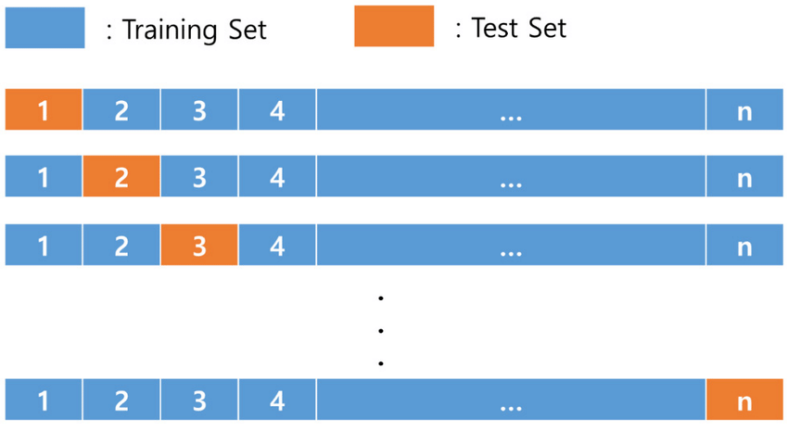


Figure 1: Leave One Out Cross Validation Diagram

variances "smoothed" with linear approximation, we then obtain the six variances (one for each bin) projected by residual variance graph. The reciprocals of the projected residual variances join to form a set of weights unlike the weights provided by variance binning.

2.3 Weight Evaluation

To find the best weights, we are interested in risk, a model's error on the test data, rather than its loss, a model's error on the training data. We utilized two methods to evaluate the weighted OPR model, both of which make judgments based on the risk produced by both the weighted and unweighted OPR models. The key difference between the two methods is *how* the risks are calculated.

For both methods, to contextualize the performance of the weighted model against the unweighted model, we take the ratio $\frac{R_{unweighted}}{R_{weighted}}$, where R is some estimate of a model's risk. This represents how much the weighted model improves on the unweighted model; since a higher risk is worse (higher error), we can interpret this ratio as a proportion.

2.3.1 Test Mean Squared Error

Mean squared error (MSE) is a measure that can adequately characterize the accuracy of a model. We calculate MSE by taking the average of the squared residual variances.

MSE only measures the error of a model applied on a fixed and complete dataset. Consequently, it becomes vulnerable to overfitting and does not evaluate models based on their ability to make predictions for unseen data. We are interested in using methods that do a better job of estimating our model's *prediction risk*, rather than just its in-sample MSE. Test-set cross validation is a method that avoids these issues by splitting data into a training and testing set; we train the model only on the training data, reserving the testing data to evaluate the model.

Combining both of these methods leads to test MSE, which is the MSE of only the test set. However, withholding a test set sacrifices a decent proportion of the complete data for training, resulting in high variance.

2.3.2 Leave-One-Out Cross Validation

Leave-One-Out Cross Validation (LOOCV) is another effective risk estimate from machine learning. Like test MSE, LOOCV splits the data into a test and training set, but instead uses only one data point as the test set. This method trains and produces a linear regression model using all but one of the matches in an FRC event (e.g. in a 160-match event, 159 matches are used for training). It then tests the trained model by predicting the omitted match's score and recording the residual. LOOCV repeats this process for each match in the event so every match is tested once. We prefer LOOCV over test MSE because it uses nearly all of the data, therefore maintaining low variance. LOOCV is detailed visually in 1 and mathematically in pages 200-203 of [7].

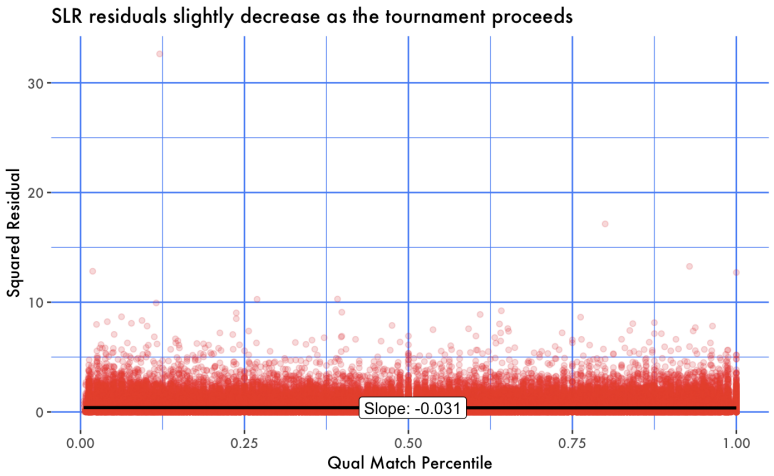


Figure 2: Simple Linear Regression on Raw OPR Residuals by Match Percentile

3 Results

3.1 Exploratory Data Analysis

Figure 2 shows the linear relationship between the qualification match percentile and the resulting squared residual. The line of best fit has a slope of -0.031 ($p < 0.001$), demonstrating that qualification matches that occur later in the tournament have smaller squared residuals. The negative slope means that on average, the error decreases as match percentile increases, which confirms a measurable relationship between match percentile and the residual produced. This supports our assertion that weighting reduces variance by determining how important each match is and assigning its appropriate relevance in calculations. Figure 2 confirms the advantage of weighting late-tournament matches more heavily by observing how they have slightly less variance overall, thus reducing error and improving predictions.

3.2 Weight Estimation

In Figure 3a, the residual variance is plotted for each of the six bins. These values represent the spread of error within each bin. The smaller this spread, the more consistent the unweighted OPR residuals. Overall, we see a parabolic trend: the residual variances are very high towards the beginning (bins 1 and 2), low towards the middle (bins 3 and 4), and increase somewhat again towards the end of the tournament (bins 5 and 6). This means that the unweighted OPR estimations are the least consistent at the extremes of a tournament, especially at the beginning. Hence, OPR should be given more weight for matches that fall in bins 3 and 4, and less weight for matches that lie further towards the beginning or end of the tournament. Linear weight smoothing produces a v-shaped variance graph with the reciprocals of the estimated weights, as shown in 3b.

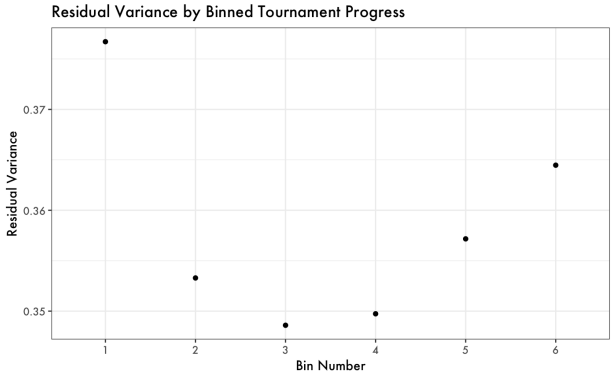
3.3 Test Mean Squared Error

Recall from section 2.3.1 that we calculate Test MSE by producing a model based solely on the training data and computing that model's MSE from the testing data. In figure 4a we plot the ratios between the unweighted and weighted event test MSEs. The mean of the distribution is 1.003, indicating that weighted OPR is .3 percent more accurate than unweighted OPR on average.

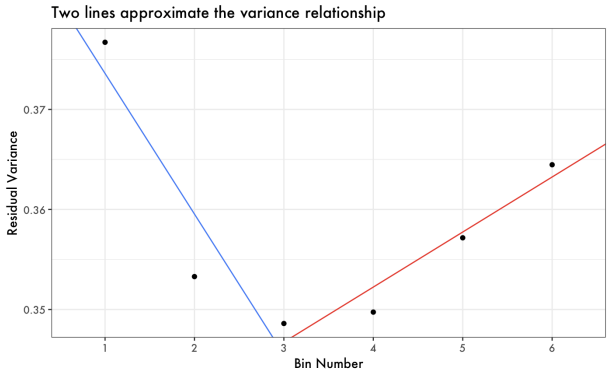
3.4 Leave-One-Out Cross Validation

An Event-Normalized LOOCV Error Ratio above 1 suggests that weighted OPR has a measurable predictive advantage over unweighted OPR. Figure 4b shows a mean LOOCV ratio of 1.001, suggesting that on average, weighting does improve OPR predictions by 0.1% as directly compared to its unweighted counterpart.

Figure 5 shows the LOOCV ratio broken down by year. In nearly all cases, the binned LOOCV ratios are larger than the linear ratios, confirming that bin weighting outperforms linearized weights. Furthermore, 2010 is the only year where the binned LOOCV ratios are below 1. Although this implies that raw OPR possesses an advantage here, there

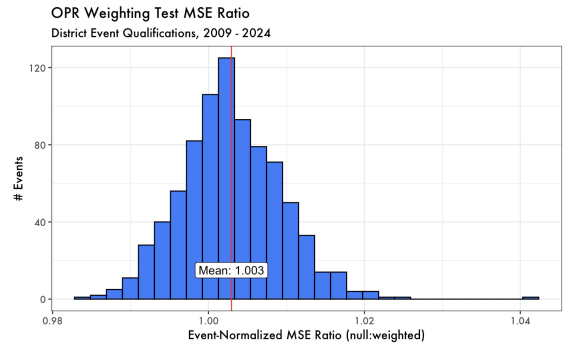


(a) Residual Variance Pattern Throughout An Event

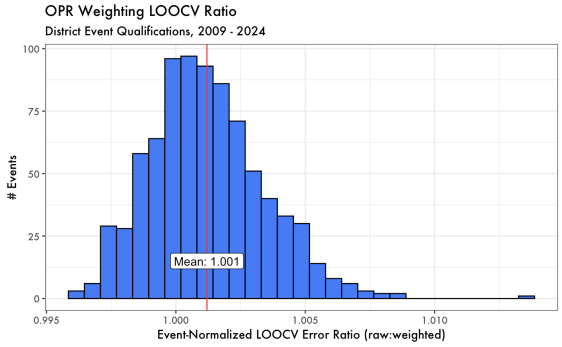


(b) Linear Approximation of Residual Variances

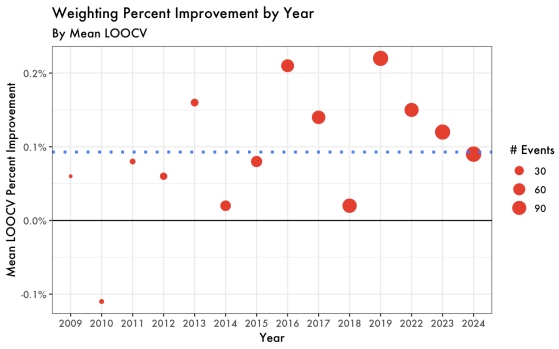
Figure 3: Residual Variance Patterns



(a) Test MSE Ratio



(b) Leave One Out Cross Validation Ratio Distribution



(c) Weighting Percent Improvement over Raw OPR

Figure 4: Weighting Evaluations

were only eight events run this year - the small sample size makes it difficult to make confidence inferences based on that year.

Figure 4c shows the percentage improvement of weighted OPR over raw OPR over time. The size of each dot scales with the number of events for that year recorded in the data, which is the number of district events played that year. The blue dotted line is the average of the data. The only year with a negative percent improvement was 2010; however, note that before 2014 no year had more than 20 district events. As the number of events grows, we see the improvement from weighting stabilize on average.

Mean LOOCV Ratios

unweighted:weighted[1]

Year	# Events	Linear	Binned
2009	7	1.0005	**1.0006**
2010	8	0.9984	**0.9989**
2011	10	0.9998	**1.0008**
2012	17	0.9997	**1.0006**
2013	19	1.0006	**1.0016**
2014	43	1.0002	**1.0002**
2015	50	0.9999	**1.0008**
2016	72	1.0021	**1.0021**
2017	81	1.0014	**1.0014**
2018	94	**1.0003**	1.0002
2019	109	**1.0023**	1.0022
2022	90	1.0014	**1.0015**
2023	109	1.0008	**1.0012**
2024	112	1.0006	**1.0009**

[1] Higher values indicate the weighting is performing better

Figure 5: LOOCV Ratios between Unweighted and Weighted OPR

4 Discussion

Independent residuals is a key assumption for linear regression. Under this assumption, linear regression is extremely efficient. However, we proved that the residuals of the OPR model are in fact not independent. Taking advantage of this invalid assumption, we can lean into the matches in which OPR is most predictive, leading to an improved model.

Weighting OPR based on match recency showed a consistent, but small improvement over unweighted OPR at district events between 2009 and 2024. Weighted Least Squares improved our estimation of teams by 0.015 Crescendo points and impacted teams' OPRs by about 0.2 points. This is not a large change, but it shows that even an unoptimized weighting can improve on unweighted OPR.

We found that the variance of unweighted OPR residuals follows a similar pattern in every year we tested 5. Notably, weighting improves OPR *independent* of OPR's value as a metric in a given year. For example, OPR did very well in 2022 and poorly in 2017, but the accuracy of OPR increased similarly with weighting for both years 4c.

4.1 Limitations

WLS is only as good as its weights. We identified *good* weightings, but did not numerically optimize to find the *best* weightings. For our first weighting model, residual variance binning, the computational cost to optimize the weighting over b bins is in $O(n^b)$. This would take around four days of computation over a reasonable grid, which was outside our scope.

To make optimization easier, we tried weight linearization (3b), which simplifies our search space. Instead of optimizing over b bins, we would optimize over two slopes and two intercepts, for $O(n^4)$. However, this simplification trades accuracy for optimization speed. In both cases, more optimization is required before we find the numerically *best* weights.

4.2 Next Steps

The next step to improving our weighting is tuning the slopes and intercepts of the piecewise linear function using cross validation. With considerable computing resources, it would also be worthwhile to brute-force optimize over the binned weights (as well as the number of bins.) This would not only find a set of best weights but would also indicate the ideal number of bins to split matches into.

In a broader scope, the primary limitation of linear regression in an FRC context is small sample size. Teams almost never play more than twelve qualification matches each in a single tournament, and while simple linear regression is an

efficient estimator [4], it doesn't stabilize until late in the tournament. To make a more useful single-event summary for FRC, we need to continue to reduce the variance, to make a model that stabilizes faster. However, per the Gauss-Markov theorem [9], we know that simple linear regression is the minimum variance unbiased linear estimator. Therefore, to improve on the variance of that estimator, you either need to accept bias, similar to how ELO models incorporate historical information, or adopt a nonlinear model. Both options provide promising avenues to improving on current regression methods in FRC.

4.3 Applications Outside the Regression Context

Any set of weights can be used to create a design matrix with repeated data entries, where the weighting vector would determine how many times a set of matches within a bin should be duplicated. Running SLR on this matrix is equivalent to running WLS on the original match matrix.

While WLS would use a set of optimized weights to compute OPR coefficients immediately, row replication would generate a design matrix with duplicated rows before applying SLR to output the *same* coefficients. Using row replication allows us to apply this data in a nonregression context and use models that incorporate higher biases to find more accurate results.

5 Technical Appendix

Code for implementing weighted least squares for OPR can be found in the scoutR repo, at `scoutR/markdown/opr_weighting`. To create the data file `district_quals_09_24.rda`, use this script.

6 Contact

For questions or for help replicating our work, email Gabriel Krotkov.

To reach out to the Girls of Steel Data Science team, email Girls of Steel.

References

[1] Scott Weingart. *'Let's do a little linear algebra'*. https://www.chiefdelphi.com/t/offense-defense-rankings-for-1043-teams/71490/19, 2006.

[2] Karthik Kanagasabapathy. Effective first strategies. In *RSN Spring Conferences presented by WPI*, 2020.

[3] Eugene Fang. The math behind opr - an introduction. *https://blog.thebluealliance.com/2017/10/05/the-math-behind-opr-an-introduction/*, 2017.

[4] Simon J. Sheather. *A Modern Approach To Regression with R*. Springer Texts in Statistics, 2009.

[5] Gabriel Krotkov. Opr as linear regression. *https://tinyurl.com/opraslinreg*, 2024.

[6] Statbotics. Evaluating frc rating models. *https://www.statbotics.io/blog/models*, 2023.

[7] Trevor Hastie Robert Tibshirani Gareth James, Daniela Witten. *An Introduction to Statistical Learning*. Springer Texts in Statistics. Springer, 2021.

[8] Gabriel Krotkov. Weighted least squares and row replication. *https://tinyurl.com/27bc6565*, 2024.

[9] Marco Taboga. Gauss markov theorem. *Lectures on probability theory and mathematical statistics. Kindle Direct Publishing. Online appendix. https://www.statlect.com/fundamentals-of-statistics/Gauss-Markov-theorem*, 2021.

[4] "efficient" meaning having the smallest possible variance for a given sample size