LS88: Sports Analytics

Regression Modeling

Linear Fit/Regression

A linear fit encodes the linear relationship between two variables

Correlation is closely linked to linear fits

➔ Small errors and higher slope → stronger correlation

Correlation quantifies the association, linear fit gives the functional relationship

We used linear fits for

➔ Estimating relationship between runs/points and wins
➔ Measuring performance of batting metrics
➔ Explanatory power of Four Factor model

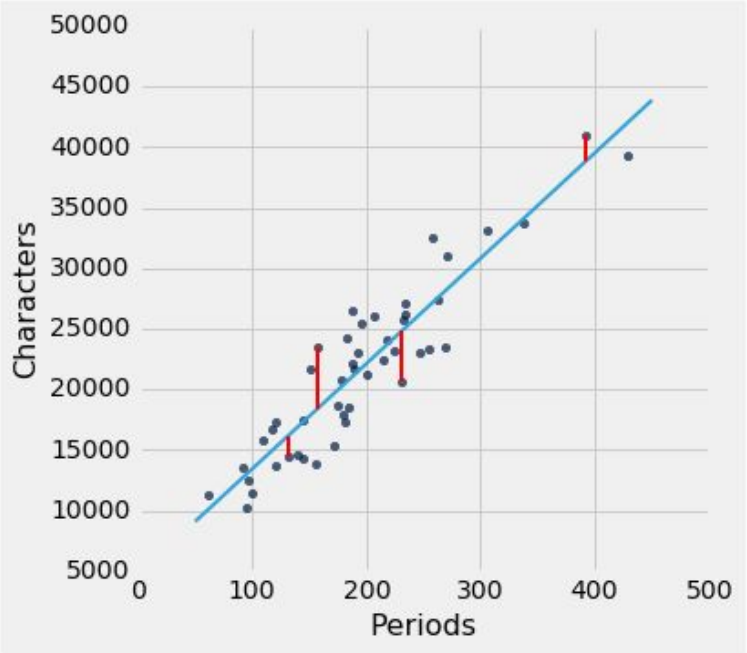
Method of Least Squares

How do we obtain the regression fit?

Method of least squares

Sec 15.3 of Inferential Thinking

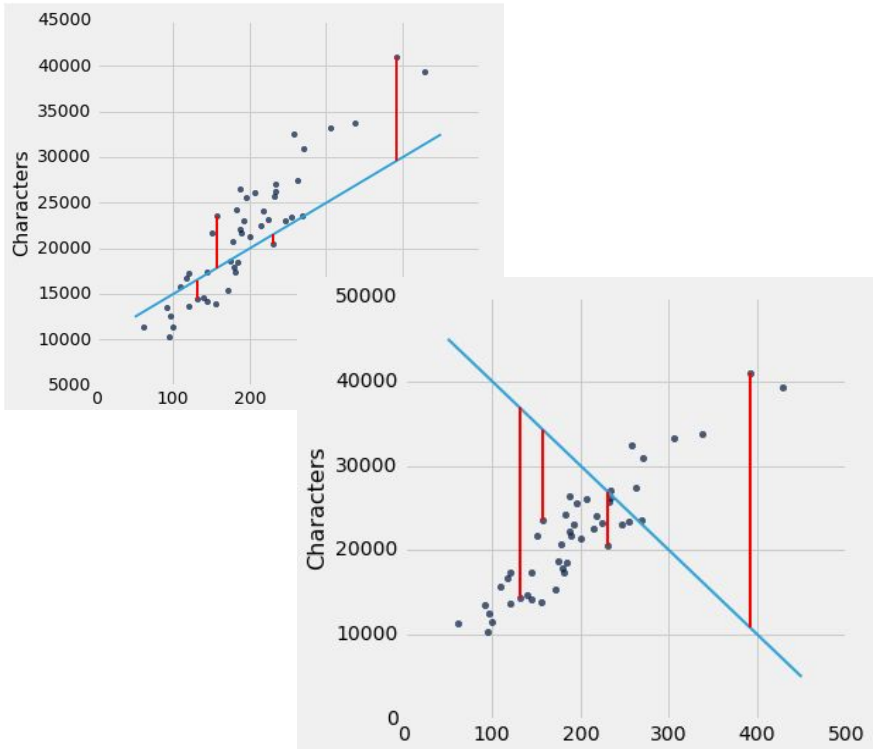
Our goal is to find a line that minimizes the Mean Squared Error (red lines)



Method of Least Squares

In trying to find the optimal fit, we'll avoid large errors and find good overall performance

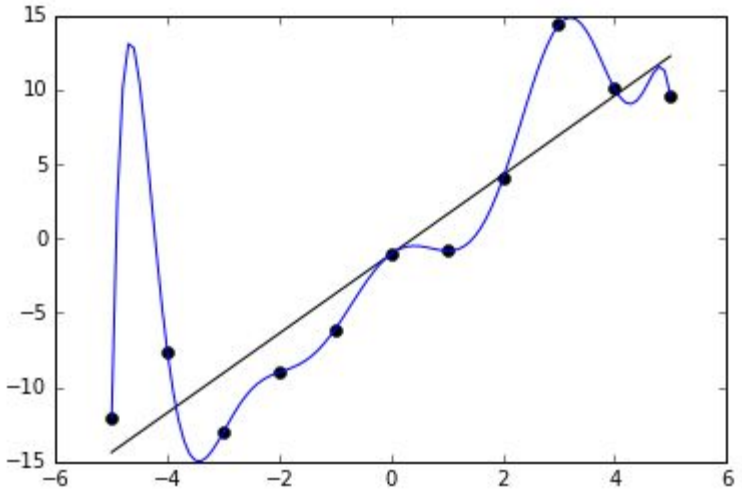
The goal is the magnitude of the errors is as small as possible



Method of Least Squares: Overfitting

After optimizing, the fit won't necessarily be flawless but we will avoid ridiculous results

As we'll see, optimizing can also backfire. We can fit the data too well leading to poor prediction on new observations



Multiple Regression

The relationship between an observation and inputs is given by:

$$\text{Observation} = \alpha + \beta_1 \cdot \text{Input}_1 + \cdots + \beta_k \cdot \text{Input}_k + \text{Error}$$

We want to quantify simultaneous impacts of variables

Realistically never enough observations to enumerate everything and compute EVs

- It worked for run values: not a lot of values, lots of data, isolated impact

Multiple Regression

Inputs can be continuous variables like height, weight, area, or anything else

We can also use discrete variables. A simple example would be a binary variable

Multiple Regression: Method of Least Squares

The relationship between an observation and inputs is given by:

$$\text{Observation} = \alpha + \beta_1 \cdot \text{Input}_1 + \cdots + \beta_k \cdot \text{Input}_k + \text{Error}$$

The method of least squares still applies

$$\text{minimize} \sum \text{Error}_i^2 = \text{minimize} \sum (\text{Observation}_i - \text{Prediction}_i)^2$$

$$\text{Prediction} = \alpha + \beta_1 \cdot \text{Input}_1 + \cdots + \beta_k \cdot \text{Input}_k$$

Multiple Regression: Interpretation

What is the prediction value?

Our expectation of the observed value given the input attributes

If we receive a new set of inputs (say we're Zillow and someone lists a house) we can use the regression model to form an expectation of the observation

Multiple Regression: Interpretation

What about the coefficients?

The marginal effect of the variable

All things being equal, if we increase a variable by 1 unit, the coefficient quantities the change in expectation

We often use continuous variables like height or weight but can also use discrete variables

A simple example would be a binary variable like “house has a jacuzzi”

Coefficient represents impact of the binary condition being true

Multiple Regression

Let’s try it with a demo

Plus/Minus

Plus/Minus

Raw Plus/Minus is a statistic that routinely gets used but is quite terrible and useless

To compute plus/minus:

- Track the score change while a player is on the court
- If the score goes up by 10, the player is a +10
- If the score goes down by 5, the player is a -5

You can compute +/- for a game or aggregate it for a season

You could normalize it to per 100 possessions if you wanted

Issues with Plus/Minus

- ➔ If you play with Lebron, you're potentially going to get rated too high
  Alternatively, if you play with someone bad, it's likely you'll be rated too low
- ➔ It also does not address opponent strength:
  A +10 against the worst team is the same as a +10 against the best team
- ➔ And if your team is good, it'll outscore opponents so every player will look positive

Plus/Minus: On/Off Differential

1. Track +/- for when a player is on versus off
2. Compute the difference (per 100 possessions)

By computing the on/off measure, you are trying to isolate the player's effect

This is a common statistical technique:

➔ Compute differences between results from a binary condition (True/False)

➔ If there is enough diversity in all the other variables (other players on the court), you will be able to measure the size of the effect

Adjusted Plus/Minus

Given that there are so many simultaneous impacts of players, this seems like a perfect opportunity to use regression modeling

Adjusted +/- uses regression to estimate the impacts and compute a +/- player rating

Top Down vs Bottom Up Metrics

Adjusted +/- is what's known as a *top down metric*

➔ Use end results (scores, wins) to estimate player performance

Top Down Metrics require methods like regression modeling

In contrast, we've done a ton with *bottom up metrics*

➔ Use collected microevents to build towards an evaluation: points, rebounds, singles, doubles, etc

Adjusted Plus/Minus

We collect data on *stints* for an entire NBA season

A *stint* is a contiguous block of time where the same 10 players are on the court together

For each stint, we track who was on the court, the number of possessions, and the offense/defense performance

We model team performance as the sum contribution of each player on the court.

Adjusted Plus/Minus

Our goal is to build this model:

$$\begin{aligned}\text{HomeTeamNetRating}_t &= \text{HomeCourtAdv} \\ &+ \text{Sum(Home Player } i\text{'s rating if player } i \text{ is in the } t\text{-th stint)} \\ &- \text{Sum(Away Player } i\text{' rating if player } i \text{ is in the } t\text{-th stint).}\end{aligned}$$

Each variable corresponds to a player

- 0 if the player *was not* on the court during the stint
- 1 if the player *was* on the court during the stint

Adjusted Plus/Minus

Our goal is to build this model:

$$\begin{aligned}\text{HomeTeamNetRating}_t &= \text{HomeCourtAdv} \\ &+ \text{Sum(Home Player } i\text{'s rating if player } i \text{ is in the } t\text{-th stint)} \\ &- \text{Sum(Away Player } i\text{' rating if player } i \text{ is in the } t\text{-th stint).}\end{aligned}$$

We end up with a table of 0s and 1s

Size is (Number of Stints) $\times$ Number of Players

Each row will have exactly 10 non-zero values

Adjusted Plus/Minus

Demo

Adjusted Plus/Minus: What can go wrong?

Regressions aren't perfect and can fail for a few simple reasons

First off, we probably should have bucketed players who didn't play much into a unified "scrub" player

A player who was in for one short stint and saw a net rating of +200 will have the regression optimized aggressively try to give him a big rating

The optimizer can easily and freely reduce the fit error doing this so why not?

Adjusted Plus/Minus: Weighted Regression

If a stint featured many possessions, the result is inherently more reliable than another stint with very few possessions

We can address this with a *weighted regression*:

$$\text{minimize} \sum \text{Weight}_i \times \text{Error}_i^2 = \text{minimize} \sum \text{Weight}_i \times (\text{Observation}_i - \text{Prediction}_i)^2$$

It's not always clear what the weights should be, but we can use number of possessions in a stint here.

Adjusted Plus/Minus: What can go wrong?

Okay, so we either dumped the “scrubs” or used a weighting.

Would that fix things?

We’d get closer but there is one issue we want to address.

Observed NBA Lineups do not behave like randomized controlled trials

➔ Sometimes two players almost always play together.

➔ Or two players always switch for each other

Adjusted Plus/Minus: What can go wrong?

In an ideal world, we'd get to randomly shuffle players around

Of course, this ignores coaches and learning to play together

So this lack of good randomization in lineups leads to a well-known issue:

Multicollinearity

The mathematical definition in a nutshell:

Given 9 players on the court, the presence of *multicollinearity* means we can do a really good job predicting the 10th player

Multicollinearity

Suppose two players almost always play together:

- ➔ If they almost always play together, the small amount of performance for when one of the players plays alone becomes magnified
- ➔ In reality, our best guess is basically to just split the performance in half
  If the pair is +5, give them each +2.5

Suppose two players always switch for each other:

- ➔ We can only reliably give the *difference* in quality between the two players
- ➔ Our best guess is to split the difference
  If the difference is 5, the better player should be +2.5 and the worse player -2.5

Multicollinearity

http://www.espn.com/nba/story/_/id/21921522/kevin-pelton-weekly-mailbag-including-wizards-playing-to-competition

Multicollinearity

I was hesitant to use this term since it's overly technical and jargony

However, Kevin Pelton does a good job of explaining:

> Readers shouldn't be scared off by the word multicollinearity, which in this context means that adjusted plus-minus has a difficult time determining which player to credit for a team's success or failure when they tend to play together frequently. I don't think that's a big issue for the Warriors because of games their stars miss due to injury and the fact that Steve Kerr tends to mix his starters and reserves. Multicollinearity is a bigger issue when coaches tend to exclusively play their starters and reserves together, or there are specific players who play only when the other is off the court.

Addressing Multicollinearity

So how do we address multicollinearity?

➔ Simultaneously minimize least squares but also *penalize aggressive fitting*

If the optimization wants to assign a big rating to someone, it better have a lot of evidence behind it, ie. the reduction in the error needs to offset the penalty

The penalty is on the sum of squares of the coefficients

$$\text{Penalty} = \text{Penalty Strength} \times \sum(\text{Player } i\text{'s Estimated Rating})^2$$

We pick a penalty parameter to quantify the strength of this penalty
There are methods that can suggest a good value.

Regularized Adjusted Plus/Minus (xRAPM)

When we fit the the penalized regression, we get Jerry Engelmann's statistic called *Regularized Adjusted Plus Minus* or xRAPM

Regularization is the same thing as penalization

- It's a term out of mathematics

Engelmann typically uses multiple years of data but we still do pretty well

xRAPM is the cousin/basis for ESPN's Real Plus/Minus (RPM) statistic

http://www.espn.com/nba/statistics/rpm/_/year/2018

RAPM Demo

RAPM: The Penalty

The penalty for each player is the same:

$$\text{Penalty} = \text{Penalty Strength} \times \sum(\text{Player } i\text{'s Estimated Rating})^2$$

The penalty is against the player's rating away from 0

So, Steph and Lonzo would be penalized the same for a rating of +8

The rating is plausible for Steph so we don't want to penalize it as much

How does ESPN's RPM handle this?

➔ Penalize Steph's rating away from a historical/projected value for Steph

RAPM: Picking the Penalty Parameter

So how do we pick the penalty parameter?

Without the penalty, our model overfits and will not perform well in prediction on new data

We don’t have new data though…

We can actually split our data into two components

- On one component, the training component, we fit the model
- On the other component, the testing component, we evaluate the model

RAPM: Picking the Penalty Parameter

Evaluating on the test component gives us an idea of how well the model performs

How do we split the data?

Randomly. And as many times as we have time for.

If we collect enough test results from these random splits, we can approximately know how our model performs in prediction

So how does this work with the penalty?

RAPM: Picking the Penalty Parameter

1. Fix the penalty parameter
2. Randomly split the data a bunch of times
3. Fit the penalized model on the data splits
4. Get the test results for each of the models and average
5. Vary the penalty parameter to see which value has the best test results

This is known as *Cross-Validation*

You cross-validate the model against data you do have so you can gauge how it will perform in prediction

Regression and Plus/Minus

RAPM is not the only game in town

APM: Adjusted Plus/Minus

Uses the basic regression. Our version was junk but with a bit of data cleaning/weighting this seemed to be fixed decently well

RAPM: Regularized Adjusted Plus/Minus

Uses penalization to help stabilize the regression. More automatic than data cleaning

SPM: Statistical Plus/Minus (might be outdated/dead)

Regress APM onto box score data (points, reb, etc). This is a two-stage process. The second stage stabilizes the results from APM

BPM: Box Plus/Minus (Basketball Reference)

Regress RAPM onto box score data. A two-stage process like SPM

Regression and Plus/Minus

And of course

RPM: Real Plus/Minus (ESPN)

- Proprietary but supposedly uses all sorts of information like aging and coaches and box score data

One last comment:

You don’t even have to do Plus/Minus for scoring. You can put in rebounding and obtain the marginal effect for rebounding