Interpretation, Hypothesis Testing and Model Fit

Jacob LaRiviere
(some content from Justin Rao)

Outline

Interpretation of Coefficients

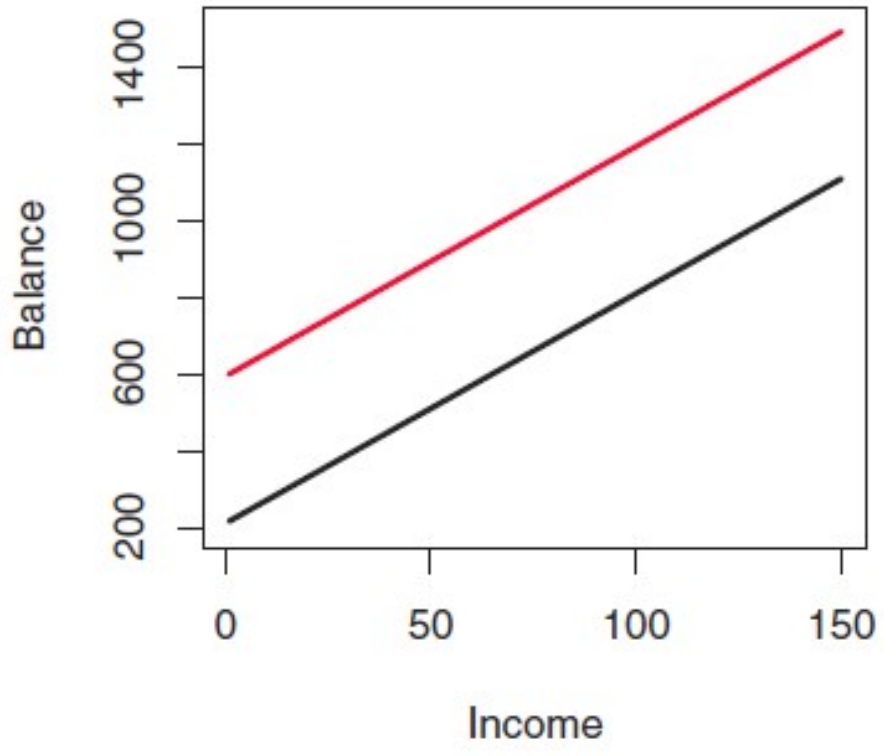
Hypothesis Testing

Model Fit

Understand binary features

β_1 allows for a different y-intercept for students

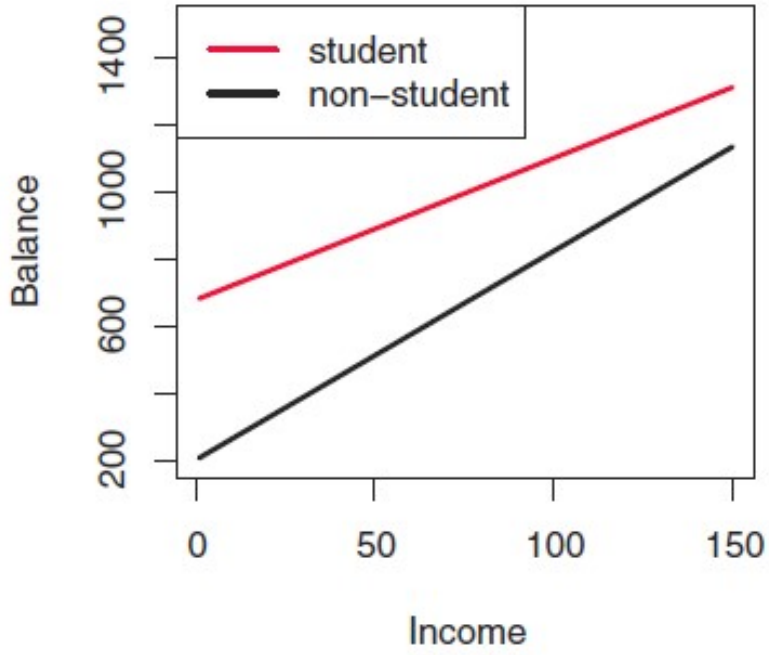
$$y_i = \alpha + \beta_1\{1\ if\ student\} + \beta_2 income + \epsilon_i$$



Understand binary features

β_1 allows for a different y-intercept for students

β_3 allows for a different Slope for students



$$y_i = \alpha + \beta_1\{1\ if\ student\} + \beta_2 income + \beta_3\{1\ if\ student\} * income + \epsilon_i$$

Interaction terms

Interaction term: multiplying two features by each other. When done as such: {continuous feature}*{binary feature}

it allows for a different slope for the group represented by the binary feature. Note, be sure to include the continuous feature without the interaction as well.

Example, suppose temperature has a different impact on # of bikeshare trips taken on weekdays vs. weekends. We can create a binary variable `is_weekend`.

If we add `is_weekend` to the model, we allow for a different baseline ridership on weekends. If we add `is_weekend*temp`, we allow temperature to have a different effect for weekends.

We can "interact" `is_weekend` with a polynomial in `temp`. This effectively gives us a new model for temperature for the weekends vs. weekdays

Conditioning on Socio-demographics

Think about two different models:

```
reg1 = glm(logmove ~ log(price)*brand, data=oj)

reg2 = glm(logmove ~ log(price)*brand + HVAL150, data=oj)
```

reg1 estimates coefficients *unconditionally* on neighborhood wealth "proxy"
reg2 estimates coefficients *conditioning* on neighborhood wealth "proxy"

NOTE 1: What if prices are set higher in richer neighborhoods and richer neighborhoods consume more OJ? **The coefficient on price is a mix of the price effect and the wealth effect!**
NOTE 2: Proxy isn't the perfect thing to measure wealth (average assets of HHs)

Interpreting regression output

`Summary(my_model)` gives the coefficient estimates (beta's), t-statistics, standard errors and p-values.

t-statistics evaluate the hypothesis that the coefficient is equal to zero. Thus t greater than 1.96 in absolute value allows us to reject this hypothesis at the 95% level.

> P-values simply convert t-stats into the probability we would get something this far from zero due to sampling chance alone.

NOTE 1: t-statistics evaluate features "one by one". Overall model fit is a better guide for explanatory power (e.g. we might be better for leaving insignificant features in sometimes)

NOTE 2: t-stats can be a good guide to throw out irrelevant features

How does this work for inference?

What about hypothesis testing? (Now assume $\alpha \neq 0$)

$$se(\hat{\beta}) = \sqrt{\hat{\sigma}^2_{\widehat{\beta}_1}/n}$$

$$\propto \frac{\frac{1}{n-2}\sum_{i=1}^{n}(x_i - \bar{x})^2\,\widehat{\epsilon_i}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\rightarrow t = \frac{\widehat{\boldsymbol{\beta}}_1 - \beta_{1,0}}{se(\hat{\beta})}$$

Both of these are themselves sample statistics from data y and X. Remember $\widehat{\boldsymbol{\beta}}_1$ is As a result, they both have their own distributions (e.g., normal and Chi Squared). The resulting ratio has the Student t distribution which is used for hypothesis testing for a given sample.

How does this work for inference?

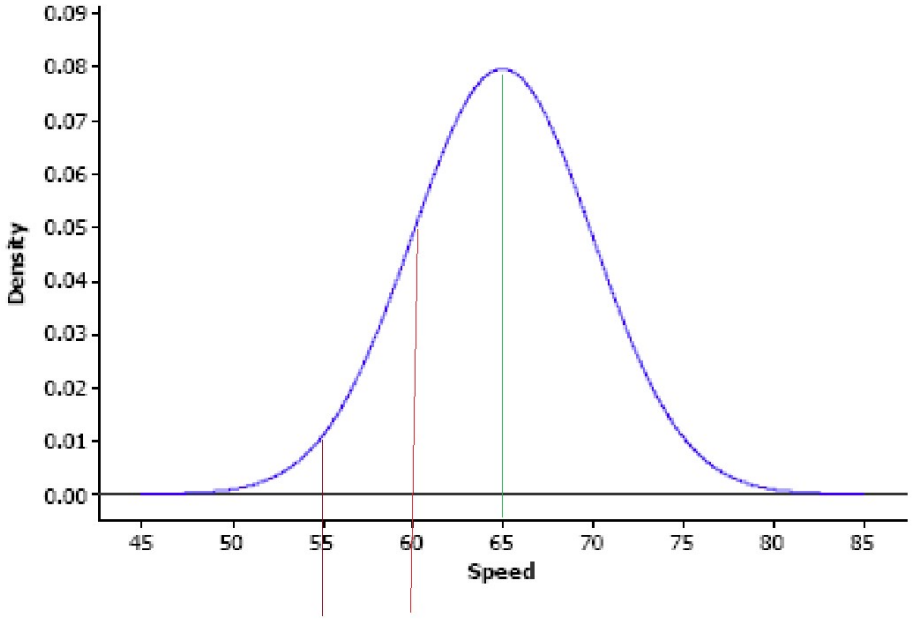
What about hypothesis testing? What is t-stat of different hypotheses given a sample size, variance in the data, and therefore the standard error?

$$H_0: \beta = 60$$
$$H_1: \beta \neq 60$$
$$\rightarrow t_{\beta=60} \approx 1$$

We fail to reject the null hypothesis that β =60 at the 5% confidence interval.

e.g., a ~30% chance that we would see sample mean speed of 65 if true mean speed were 60.

NOTE: that 30% is the p-value



$t_{\beta=55} \approx 2 \quad t_{\beta=60} \approx 1$

$$H_0: \beta = 55$$
$$H_1: \beta \neq 55$$
$$\rightarrow t_{\beta=55} \approx 2$$

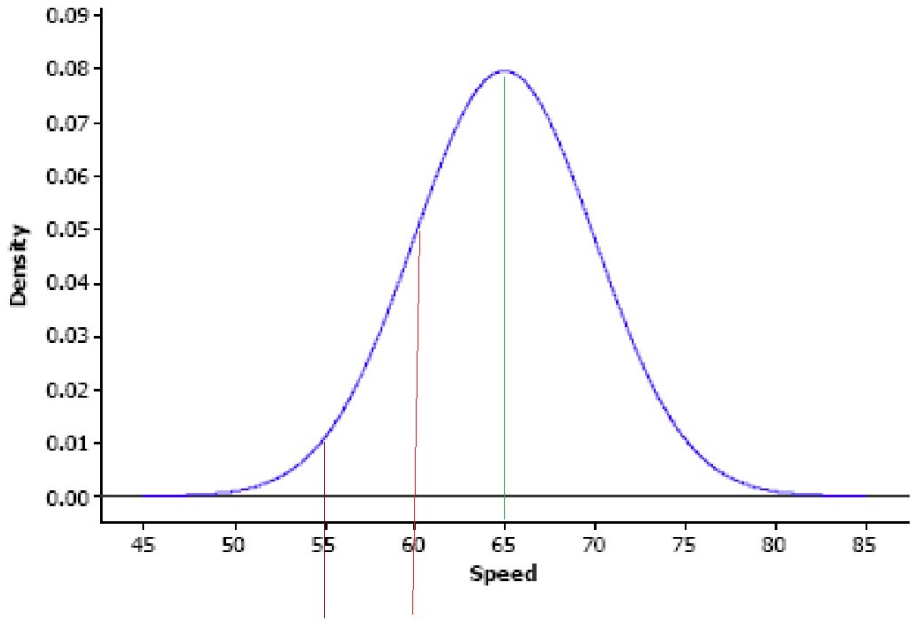
We reject the null hypothesis that $\beta = 55$ at the 5% confidence interval.

e.g., only a 5% chance that we would see sample mean speed of 65 if true mean speed were 55.

NOTE: that 5% is the p-value

How does this work for inference?

What about hypothesis testing? What is t-stat of different hypotheses given a sample size, variance in the data, and therefore the standard error?



$t_{\beta=55} \approx 2 \quad t_{\beta=60} \approx 1$

Words to the wise:

1) If you have a really small data set the standard errors will be big. Thus its hard to reject hypothesis.
2) If you test a bunch of hypotheses, things get more complicated.
3) This is about hypothesis testing *not* model prediction.

How does this work for inference?

Lets test a hypothesis:

Does price sensitivity increase as neighborhoods increase in wealth?

Model Fit

Hypothesis Testing versus Prediction

In certain scientific applications we often want to do hypothesis testing

In others we might care about prediction. These are often "researcher as engineer" scenarios.

e.g., Kleinberg et. al. 2017.
Who reoffends while out on bail?

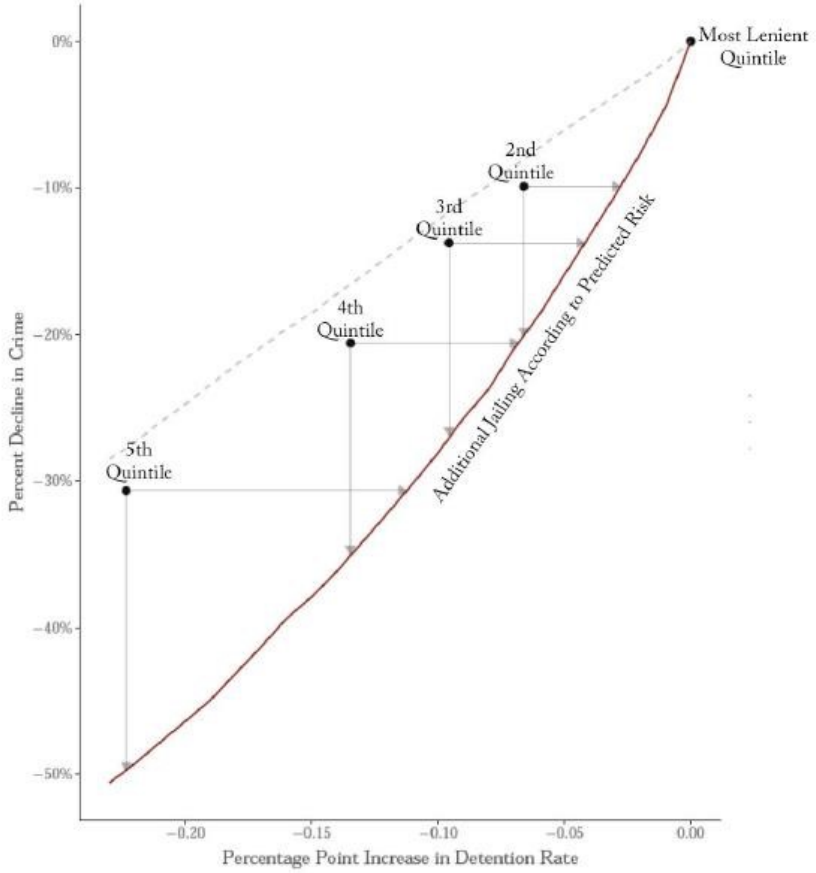


Figure 5: Comparing Impact of Detaining in order of Predicted Risk To What Judges Achieve

Feature explosion

If we have two raw features, A and B, how many models can we make?

Without interactions (4): none, just A, just B, A + B
With interactions (8): A*B, A+B + A*B, A + A*B, B+ A*B

If we have ***p*** features then we have 2^{p+1} possible models!

$$p = 29 \longrightarrow 1{,}073{,}741{,}824$$

We cannot possible run all these models… so we'll learn methods to guide us (e.g., machine learning and cross validation)

Human intelligence is often a great guide as well.

Cross Validation & Model Selection

If we have two raw features, A and B, how many models can we make?

Without interactions (4): none, just A, just B, A + B
With interactions (8): A*B, A+B + A*B, A + A*B, B+ A*B

If we have ***p*** features then we have 2^{p+1} possible models!

$$p = 29 \longrightarrow 1{,}073{,}741{,}824$$

We cannot possible run all these models… so we'll learn methods to guide us (e.g., machine learning and cross validation)

Human intelligence is often a great guide as well.

Validation set methodology

Train the model with a subset of the data

Test the model on the remaining data (the *validation set*)

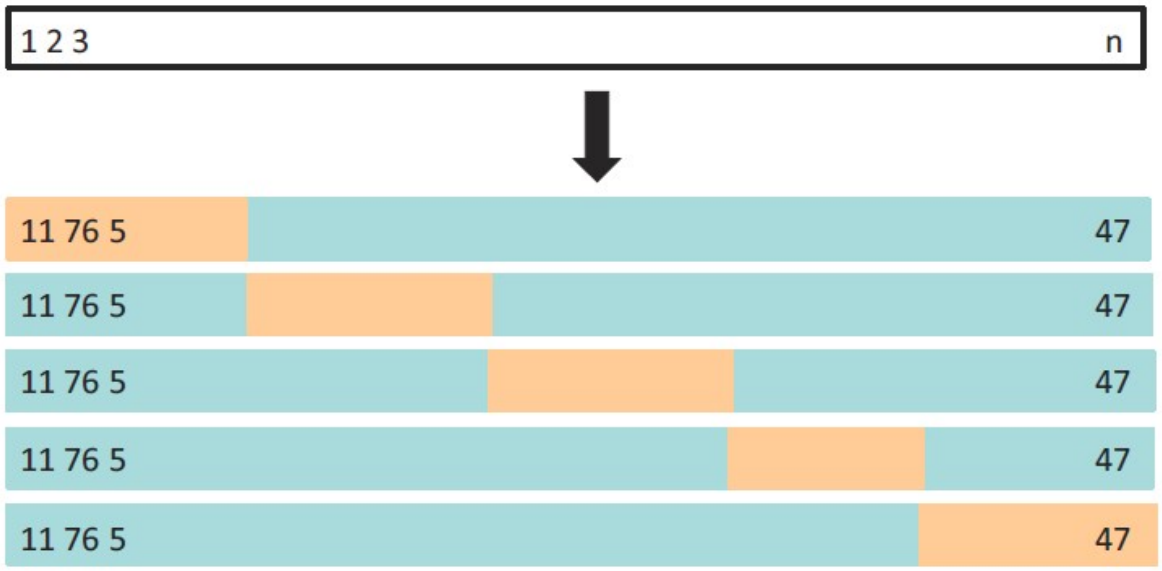
What data to choose for training vs. test?

In a time-series dimension, it is natural to hold out the last year (or time period) of the data, to simulate predicting the future based on all past data. In most settings, however, we'll randomly select our training/test sets.

Cross-validation

A class of methods to do many training/test splits and average over all the runs

Here is a simple example of 5-fold cross validation. Gives 5 test sets → 5 estimates of MSE.
The 5-fold CV estimate is obtained by averaging these values.



K-fold cross-validation

Split the data up into K "folds". Iteratively leave fold k out of the training data and use it to test.

The more folds, the smaller each testing set is (more training data), but the more times we need to run the estimation procedure. Using rules of thumb like 5—10 folds is often utilized in practice. This can be done with a simple for loop in R

For generalized linear models, the `cv.glm()` function can be used to perform k-fold cross validation. For example, this code loops over 10 possible polynomial orders and computes the 10-fold cross-validated error in each step

```
> set.seed(17)
> cv.error.10=rep(0,10)
> for (i in 1:10){
+ glm.fit=glm(mpg~poly(horsepower,i),data=Auto)
+ cv.error.10[i]=cv.glm(Auto,glm.fit,K=10)$delta[1]
+ }
> cv.error.10
[1] 24.21 19.19 19.31 19.34 18.88 19.02 18.90 19.71 18.95 19.50
```

K-fold cross-validation

Split the data up into K “folds”. Iteratively leave fold k out of the training data and use it to test.

Lets do an example with the Orange Juice data:

> Should we include socio-demographic characteristics or not if we care about predicting sales and order inventory.

Parametric versus Non-parametric Models

Parametric vs. non-parametric regression

$Y = f(X) + \epsilon$

Parametric: f defined by a model we write down with *parameters that we have to estimate* (e.g. $\alpha, \beta_1, etc.$)

Non-parametric: we directly fit the data

Local averaging

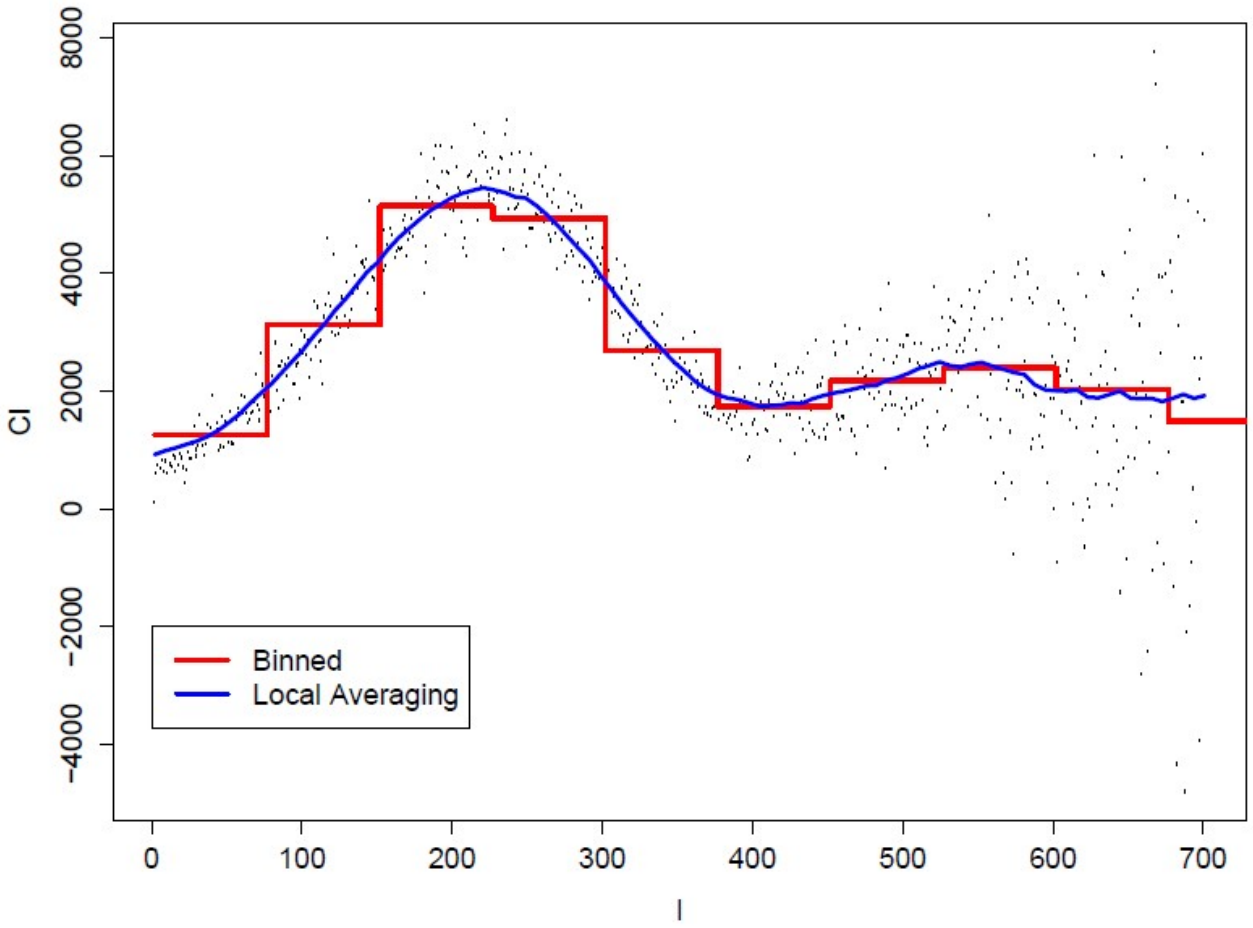
For each x let

$$N_x = \left[x - \frac{h}{2},\ x + \frac{h}{2}\right].$$

This is a moving window of length h, centered at x. Define

$$\widehat{f}(x) = \text{mean}\Big\{Y_i :\ X_i \in N_x\Big\}.$$

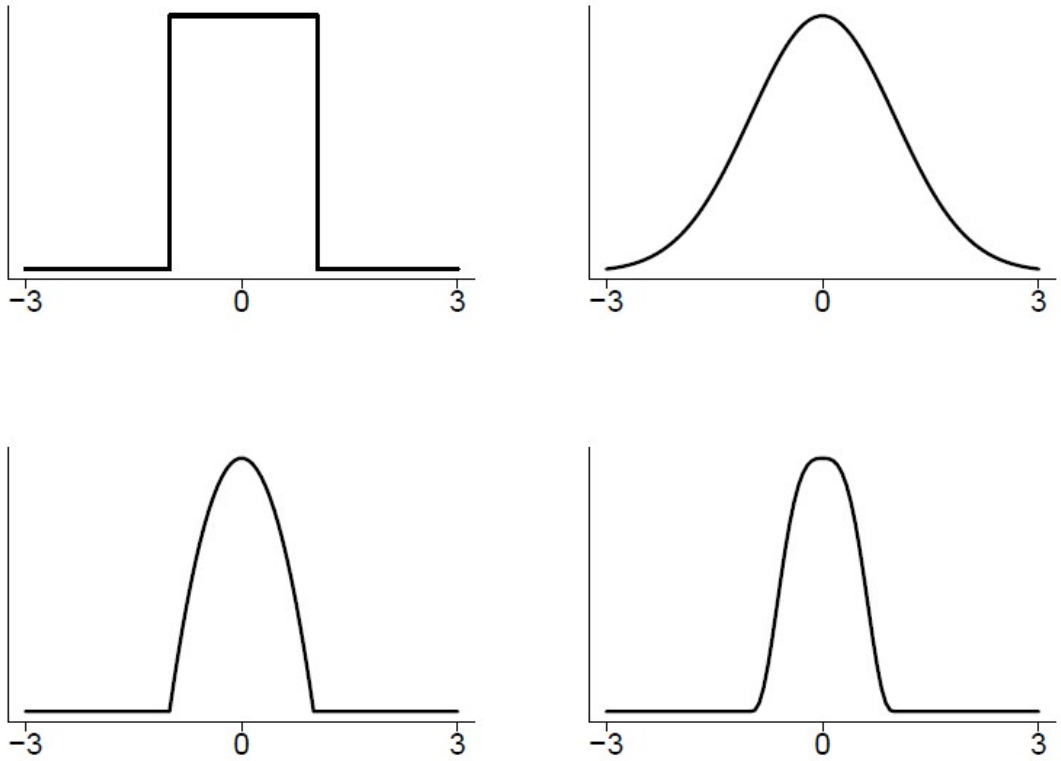
Local averaging vs. bins



Kernels

A method of assigning higher weight to the points closer to the target point I am trying to fit.

Choice of kernel usually does not matter much.



Bandwidths

Bandwidths tell us how wide to make our kernel (window)

Larger bandwidths → smoother functions because there is more data used to make the averages

Bandwidth is sometimes called the smoothing parameter.

Bandwidths matter!

K-nearest neighbors

An alternative to bandwidths is instead of specifying a fixed window, we can say “use the nearest K nearest neighbors”

This ensures each bin has the same amount of data and can be a useful tool

Functionally it will often be quite similar to using a bandwidth, except when there are “sparse data” issues (then K-NNN is preferred)

Often expressed as a fraction of data. E.g. NN=.1 says “each bin is 10% of data”

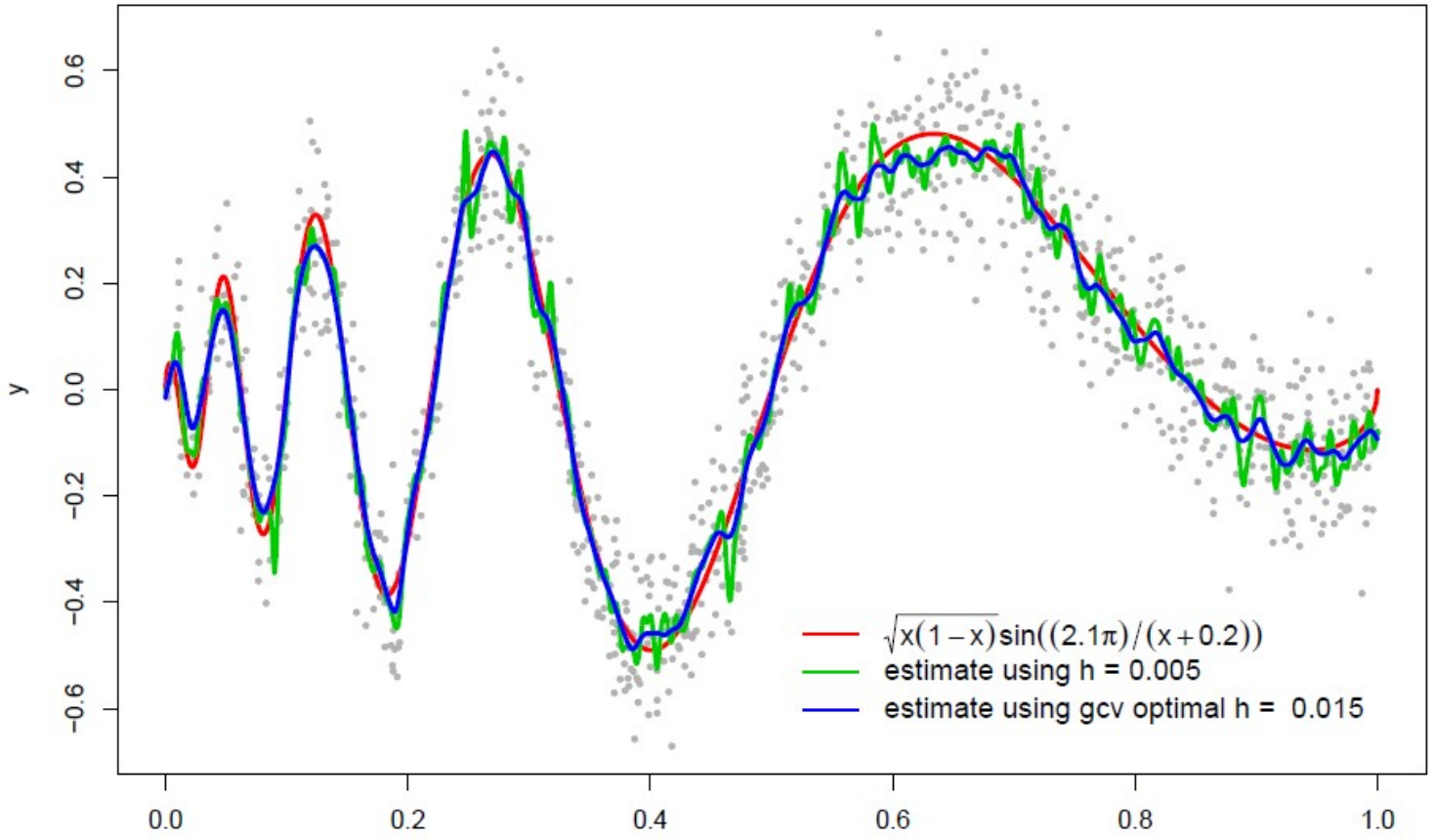
Local polynomial regression

Thing of local averaging as fitting a constant for a given window of data

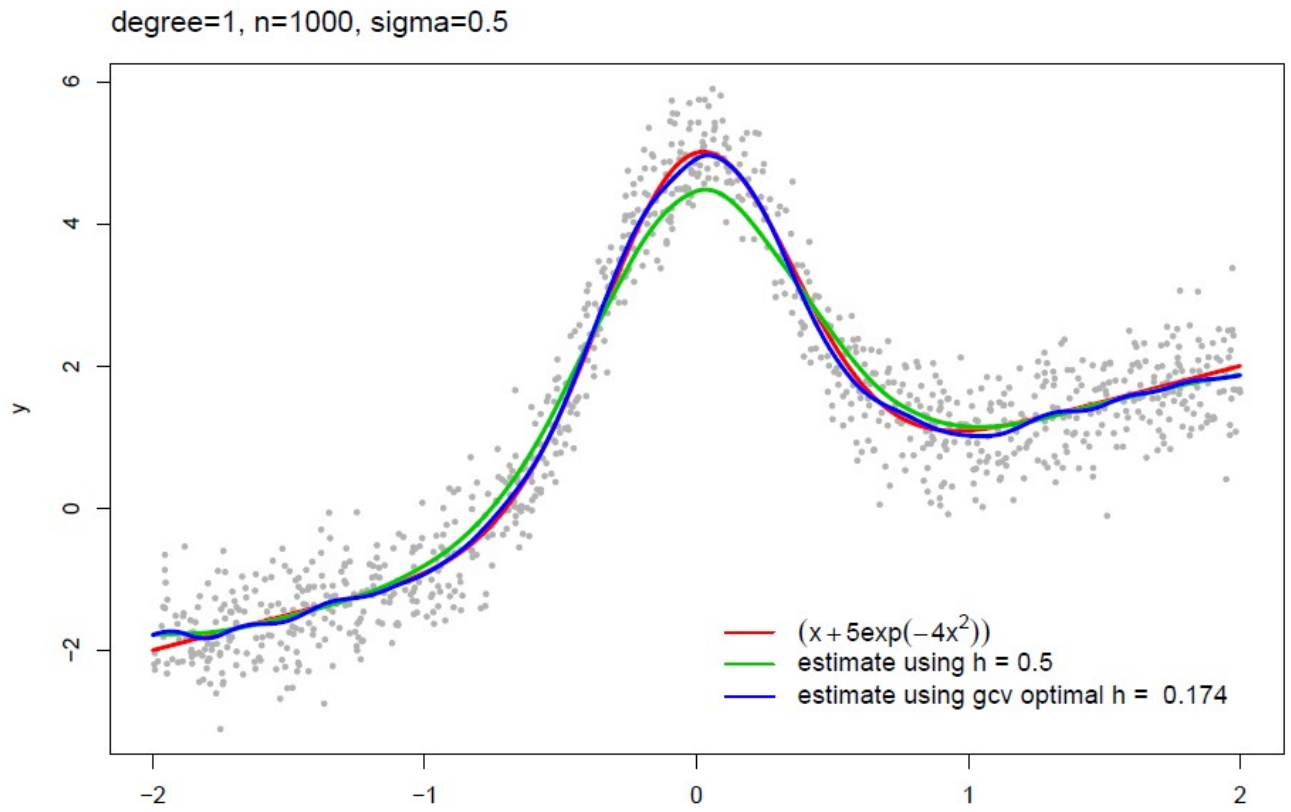
Instead of fitting a constant, we can fit a line (y=a*x + b) or a polynomial (y=a*x + b*x^2 + c), etc.

It is thus more general than local averaging, but very similar

Some examples



Some examples



Local polynomial in R `locfit`

```
install.packages(locfit)
library(locfit)
```

Modeler needs to specify: bandwidth (or fraction of nearest neighbors), degree of the local polynomial

Contrast: in parametric regression, we had to write down a model

Aside: we'll learn that non-parametric methods struggle in higher dimensions

When to use each model?

If there are a relatively small number of features (e.g. <4) then non-parametric makes a lot of sense.

With many features, the "curse of dimensionality" sets in and non-parametric methods fall apart (the "neighborhood" is generally empty)

Parametric models using interaction terms and polynomials are the preferred method with many features

"Semi-parametric" methods combine elements of both