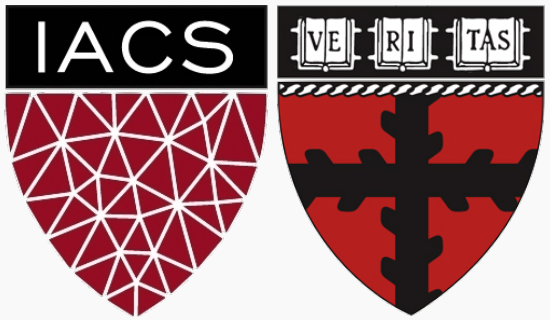
Lecture #4: Introduction to Regression

Data Science 1
CS 109A, STAT 121A, AC 209A, E-109A

Pavlos Protopapas Kevin Rader
Margo Levine Rahul Dave



Lecture Outline

Announcements

Data

Statistical Modeling

Regression vs. Classification

Error, Loss Functions

Model I: k-Nearest Neighbors

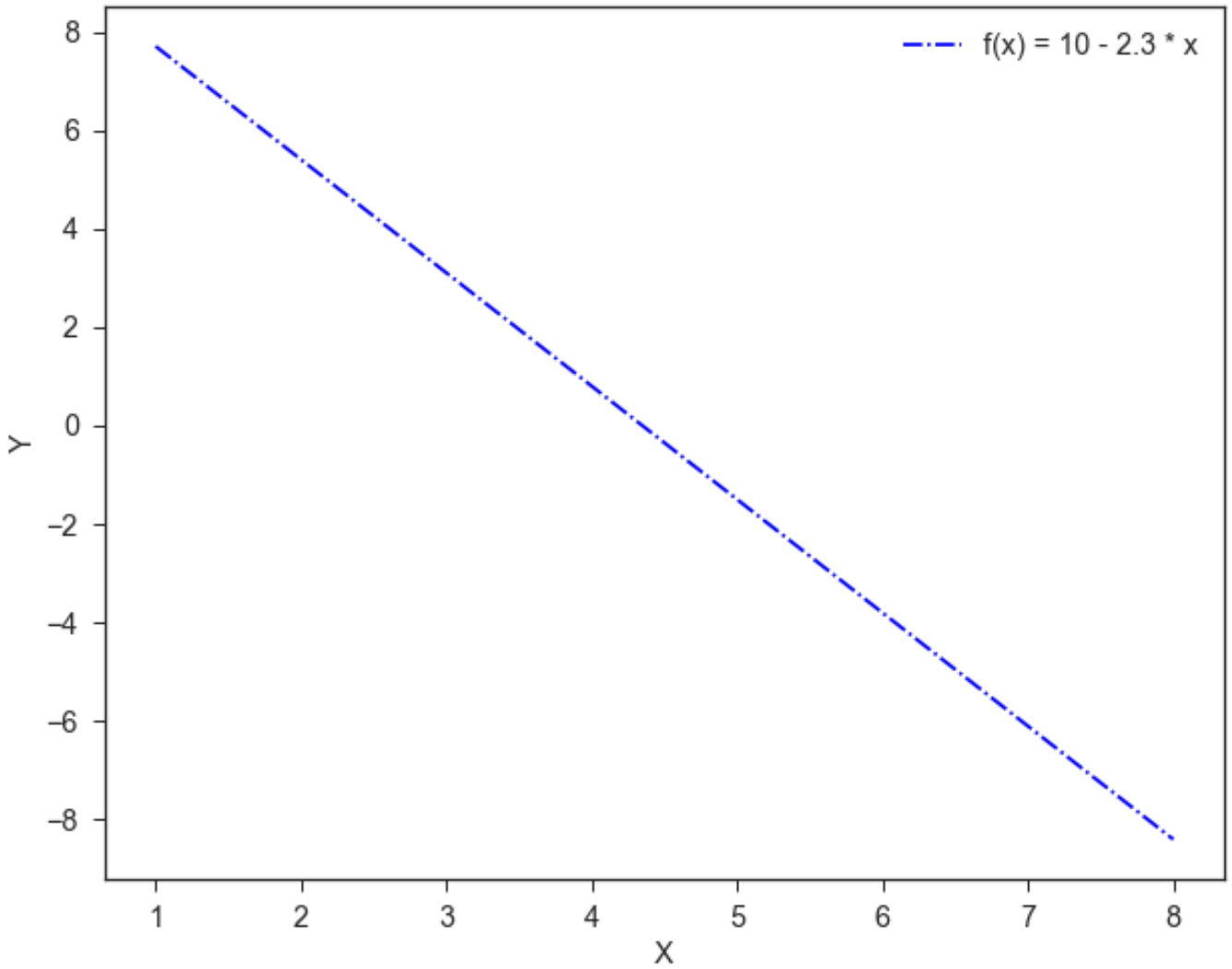
Model II: Linear Regression

Evaluating Model

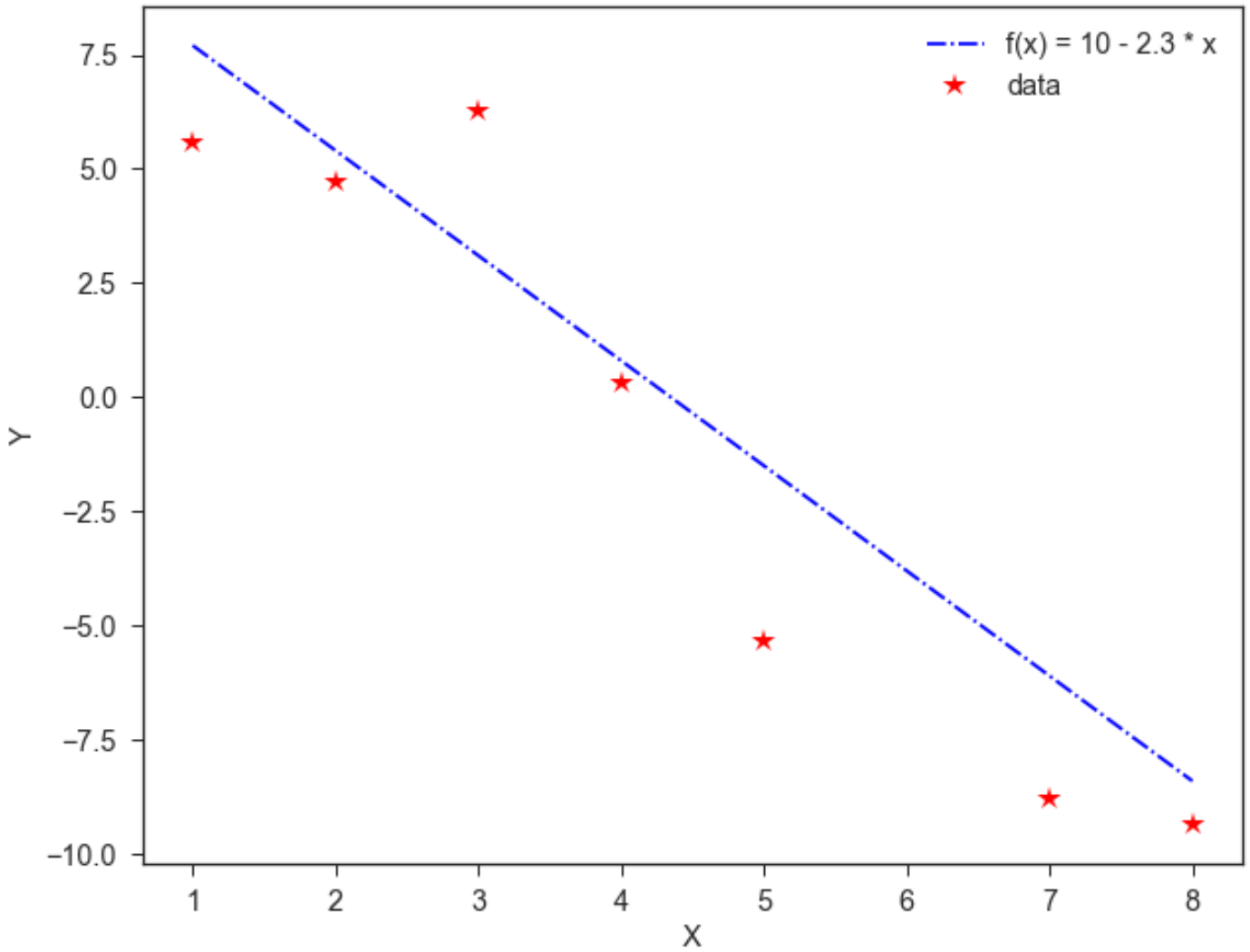
Comparison of Two Models

Model II: Linear Regression

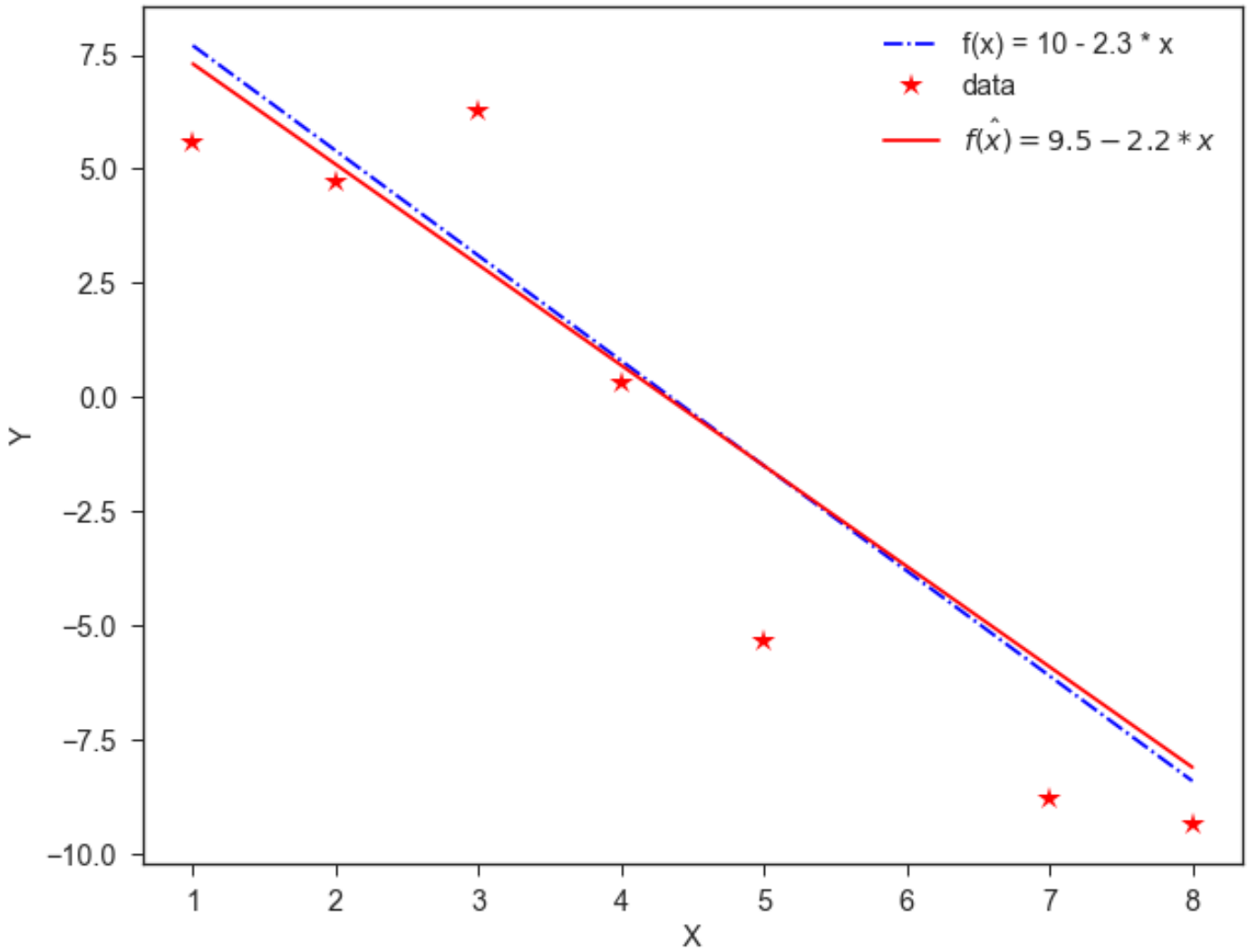
Linear Models in One Variable



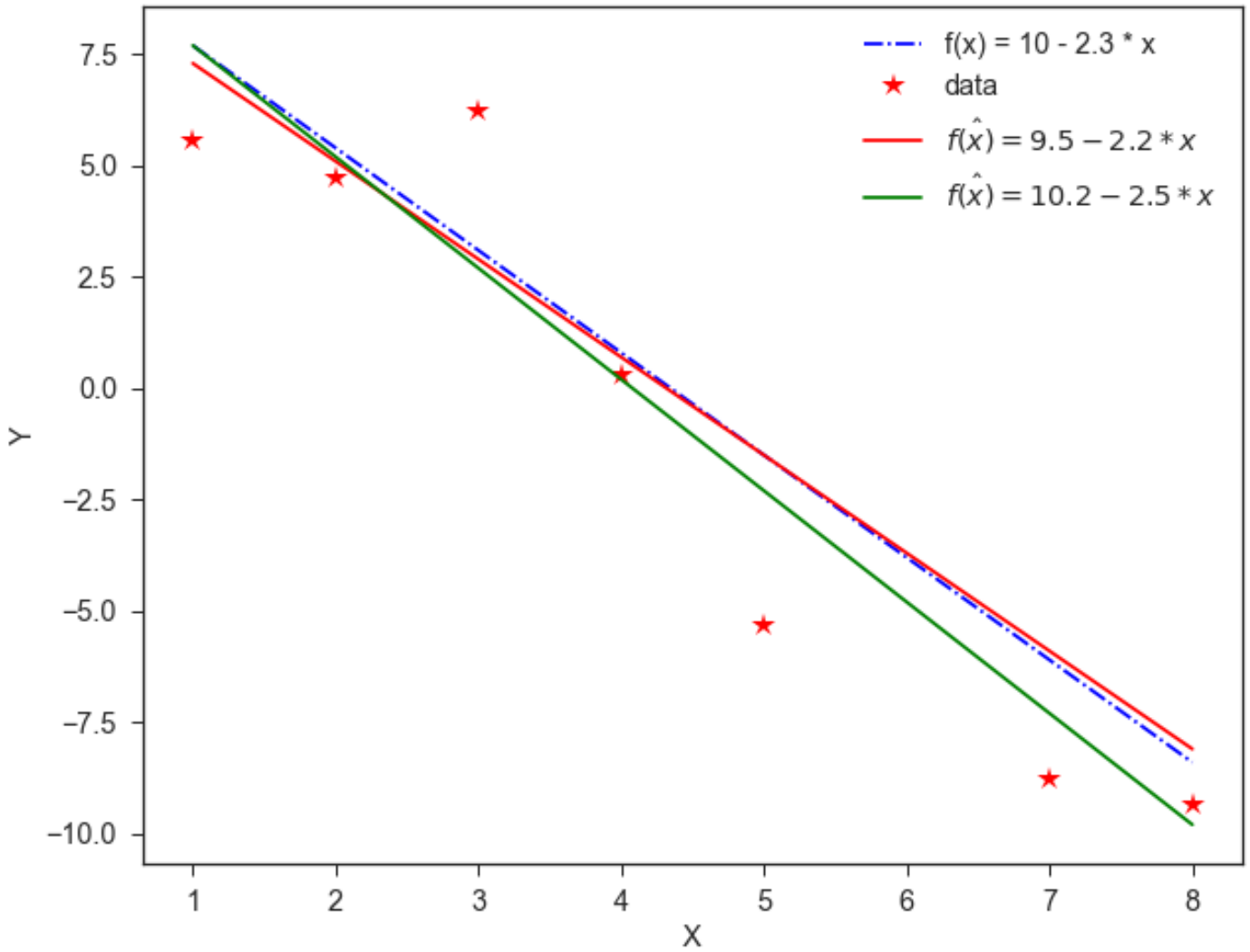
Linear Models in One Variable



Linear Models in One Variable



Linear Models in One Variable



Linear Models in One Variable

Note that in building our kNN model for prediction, we did not compute a closed form for $\widehat{f}$, our estimate of the function, f, relating predictor to response.

Alternatively, if each observation has only one predictor, we can build a model by first assuming a simple form for f (and hence $\widehat{f}$), say a ***linear form***,

$$Y = f(X) + \epsilon = \beta_1 X + \beta_0 + \epsilon.$$

Again, ϵ is the random quantity or ***noise*** by which observed values of Y differ from the rule $f(X)$.

Inference for Linear Regression

If our statistical model is

$$Y = f(X) + \epsilon = \beta_1^{\text{true}} X + \beta_0^{\text{true}} + \epsilon,$$

then it follows that our estimate is

$$\widehat{Y} = \widehat{f}(X) = \widehat{\beta}_1 X + \widehat{\beta}_0$$

where $\widehat{\beta}_1$ and $\widehat{\beta}_0$ are estimates of β_1 and β_0, respectively, that we compute using observations.

Recall that our intuition says to choose $\widehat{\beta}_1$ and $\widehat{\beta}_0$ in order to minimize the predictive errors made by our model, i.e. minimize our loss function.

Inference for Linear Regression

Again we use MSE as our loss function,

$$L(\beta_0, \beta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - (\beta_1 X + \beta_0)\right]^2.$$

Then the optimal values for $\widehat{\beta}_1$ and $\widehat{\beta}_0$ should be

$$\widehat{\beta}_0, \widehat{\beta}_1 = \underset{\beta_0, \beta_1}{\text{argmin}} L(\beta_0, \beta_1).$$

Now, taking the partial derivatives of L and finding the global minimum will give us explicit formulae for $\widehat{\beta}_0, \widehat{\beta}_1$,

$$\widehat{\beta}_1 = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}, \qquad \widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x}$$

where $\overline{y}$ and $\overline{x}$ are sample means. The line $\widehat{Y} = \widehat{\beta}_1 X + \widehat{\beta}_0$ is called the ***regression line***.

Linear Regression: A Simple Example

Recall our simple example from before, where we observe the average cab fare in NYC using the time of day,

X	1	2	3	4	5
Y	6	7	4	3	2

By our formula, we compute the regression line to be

$$\widehat{Y} = -1.2X + 8$$

Using this model, we can generate predicted responses:

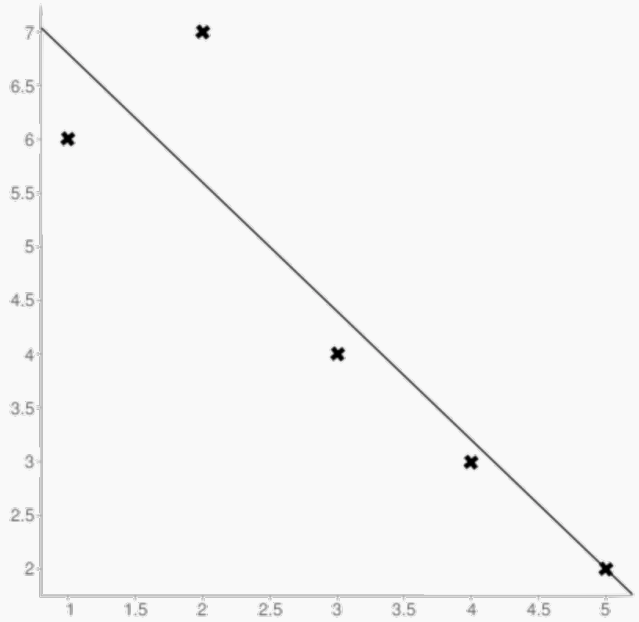
$$\widehat{Y} = (6.8, 5.6, 4.4, 3.2, 2.0)$$

Let's graph our linear model against the observations.

Linear Regression: A Simple Example

Why doesn't our line fit the observations exactly? There are two possibilities:

- f is not a linear function
- the difference between prediction and observation is due to the noise term in $Y = f(X) + \epsilon$.



Regardless of the form of f, the presence of the random term ϵ means that the predictions made using $\widehat{f}$ will never exactly match the observations.

Question: Is it possible to measure how confidently $\widehat{\beta}_0$, $\widehat{\beta}_1$ approximate the true parameters of f?

Evaluating Model

Evaluating Model Things to Consider

- How well do we know $\hat{f}$?
  The confidence intervals of our $\hat{\beta}_0$ and $\hat{\beta}_1$?
- Evaluating Significance of Predictors
  Does the outcome depend on the predictors?
- Model Fitness
  How does the model perform predicting?
- Comparison of Two Models
  How do we choose from two different models?

Understanding Model Uncertainty

We interpret the ϵ term in our observation

$$Y = f(X) + \epsilon$$

to be noise introduced by random variations in natural systems or imprecisions of our scientific instruments.

We call ϵ the measurement error or ***irreducible error***. Since even predictions made with the actual function f will not match observed values of Y.

Due to ϵ, every time we measure the response Y for a fix value of X we will obtain a different observation, and hence a different estimate of β_0 and β_1.

Uncertainty In $\hat{\beta}_0$ and $\hat{\beta}_1$

Again due to ϵ, if we make only a few observations, the noise in the observed values of Y will have a large impact on our estimate of β_0 and β_1.

If we make many observations, the noise in the observed values of Y will 'cancel out'; noise that biases some observations towards higher values will be canceled by the noise that biases other observations towards lower values.

This feels intuitively true but requires some assumptions on ϵ and a formal justification - or at least an example.

Uncertainty In $\hat{\beta}_0$ and $\hat{\beta}_1$

In summary, the variations in $\widehat{\beta}_0$, $\widehat{\beta}_1$ (estimates of β_0 and β_1 respectively) are affected by

- (**Measurement**) $\text{Var}[\epsilon]$, the variance (the scale of the variation) in the noise, ϵ
- (**Sampling**) n, the number of observations we make

The variances of $\widehat{\beta}_0$, $\widehat{\beta}_1$ are also called ***standard errors***, which we will see later.

Bootstrapping for Estimating Sampling Error

With some assumption on ϵ, we can compute the variances or standard errors of $\widehat{\beta}_0$ and $\widehat{\beta}_1$ analytically.

The standard errors can also be estimated empirically through ***bootstrapping***.

Definition

Bootstrapping is the practice of estimating properties of an estimator by measuring those properties by, for example, sampling from the observed data.

For example, we can compute $\widehat{\beta}_0$ and $\widehat{\beta}_1$ multiple times by randomly sampling from our data set. We then use the variance of our multiple estimates to approximate the true variance of $\widehat{\beta}_0$ and $\widehat{\beta}_1$.

Comparison of Two Models

Parametric vs. Non-parametric Models

Linear regression is an example of ***a parametric model***, that is, it is a model with a fixed form and a fixed number of parameters that does not depend on the number of observations in the training set.

kNN is an example of a ***non-parametric model***, that is, it is a model whose structure depends on the data. The set of parameters of the kNN model is the entire training set.

In particular, the number of parameters in kNN depends on the number of observations in the training set.

kNN vs. Linear Regression

So which model is better? Rather than answer this question, let's define 'better'.

To compare two models, we can consider any combination of the following criteria (and possibly more):

- Which model gives less predictive error, with respect to a loss function?
- Which model takes less space to store?
- Which model takes less time to train (perform inference)?
- Which model takes less time to make a prediction?

Bibliography

1. Bolelli, L., Ertekin, S., and Giles, C. L. ***Topic and trend detection in text collections using latent dirichlet allocation***. In European Conference on Information Retrieval (2009), Springer, pp. 776-780.
2. Chen, W., Wang, Y., and Yang, S. ***Efficient influence maximization in social networks. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (2009)***, ACM, pp. 199-208.
3. Chong, W., Blei, D., and Li, F.-F. ***Simultaneous image classification and annotation. In Computer Vision and Pattern Recognition***, 2009. CVPR 2009. IEEE Conference on (2009), IEEE, pp. 1903-1910.
4. Du, L., Ren, L., Carin, L., and Dunson, D. B. ***A bayesian model for simultaneous image clustering, annotation and object segmentation***. In Advances in neural information processing systems (2009), pp. 486-494.
5. Elango, P. K., and Jayaraman, K. ***Clustering images using the latent dirichlet allocation model.***
6. Feng, Y., and Lapata, M. ***Topic models for image annotation and text illustration***. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (2010), Association for Computational Linguistics, pp. 831-839.
7. Hannah, L. A., and Wallach, H. M. ***Summarizing topics: From word lists to phrases.***
8. Lu, R., and Yang, Q. ***Trend analysis of news topics on twitter***. International Journal of Machine Learning and Computing 2, 3 (2012), 327.