Lecture 12: Linear Regression I & Gradient Descent

COSC 480 Data Science, Spring 2017

Michael Hay

Logistics

- Lab 4 out soon (late tonight)
- Textbook…
- Quiz tomorrow

Let's focus on modeling a *single* predictor variable

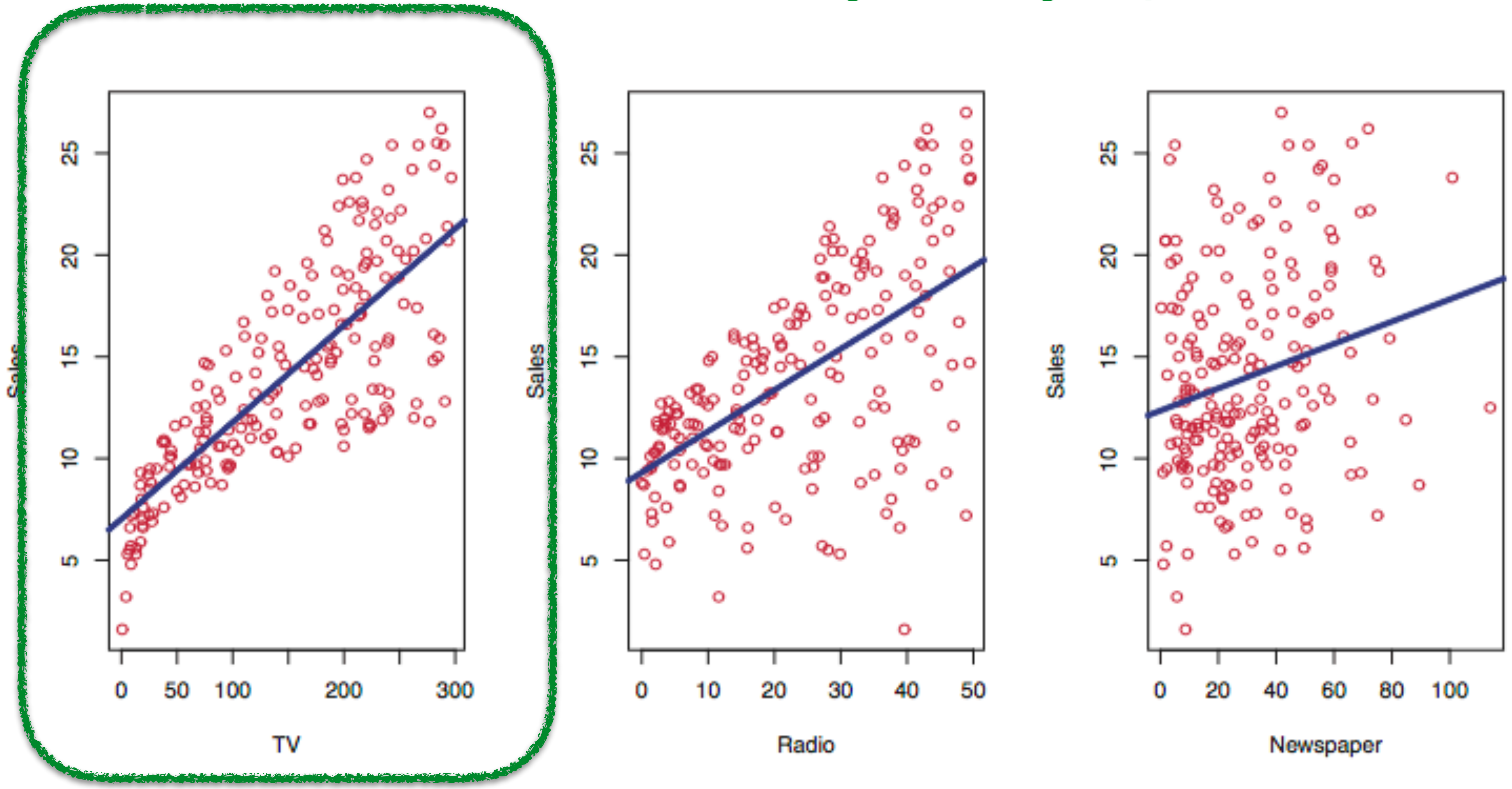

FIGURE 2.1. *The* Advertising *data set. The plot displays* sales*, in thousands of units, as a function of* TV*,* radio*, and* newspaper *budgets, in thousands of dollars, for 200 different markets. In each plot we show the simple least squares fit of* sales *to that variable, as described in Chapter 3. In other words, each blue line represents a simple model that can be used to predict* sales *using* TV*,* radio*, and* newspaper*, respectively.*

Credit: http://www-bcf.usc.edu/~gareth/ISL/

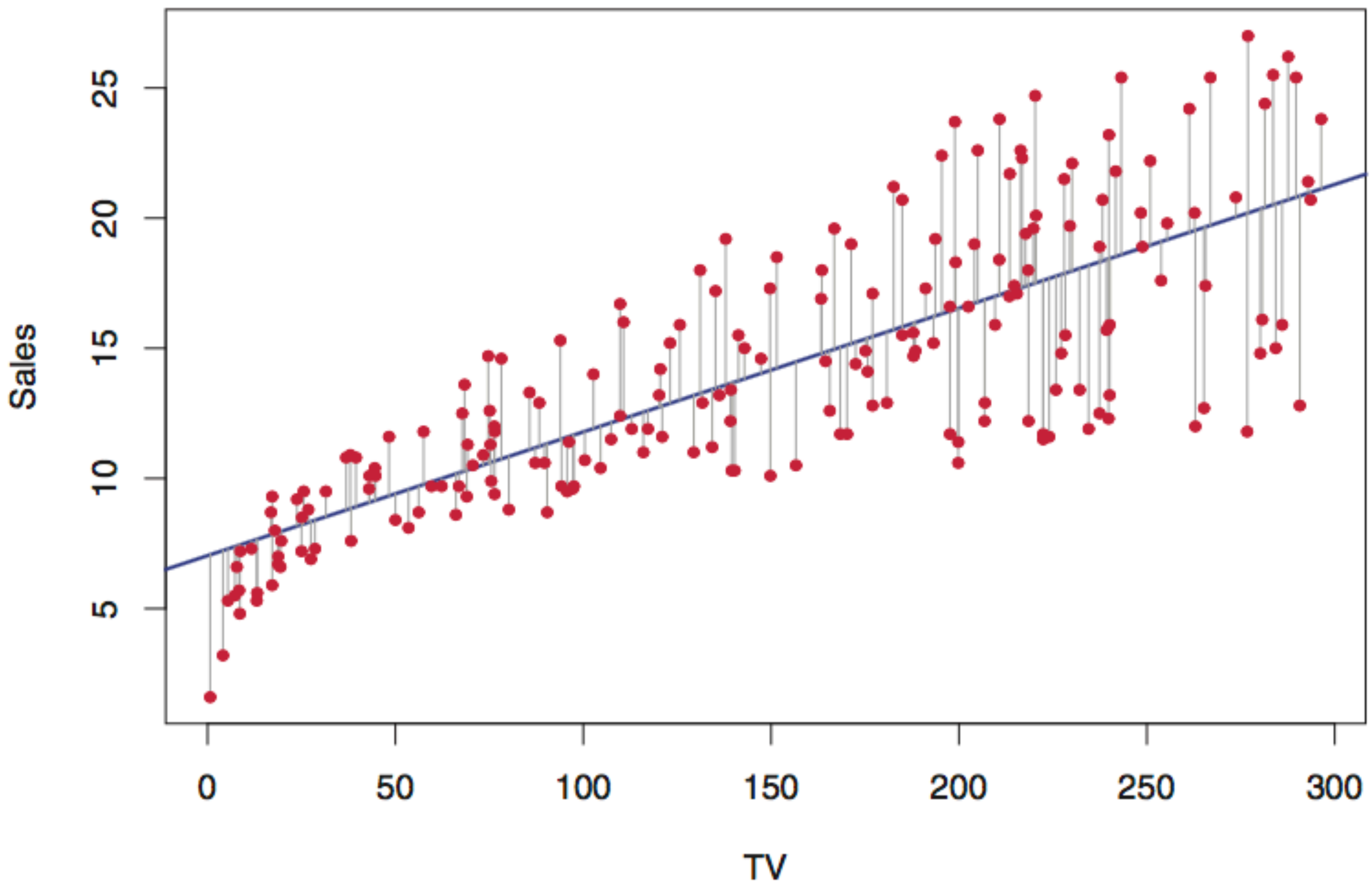

FIGURE 3.1. *For the* `Advertising` *data, the least squares fit for the regression of* `sales` *onto* `TV` *is shown. The fit is found by minimizing the sum of squared errors. Each grey line segment represents an error, and the fit makes a compromise by averaging their squares. In this case a linear fit captures the essence of the relationship, although it is somewhat deficient in the left of the plot.*

Credit: http://www-bcf.usc.edu/~gareth/ISL/

Algorithm 1 Gradient descent for simple linear regression

procedure GRADIENTDESCENT(cost function J, step size η, tolerance t)
    Initialize β_0, β_1 arbitrarily.
    cost $\leftarrow J(\beta_0, \beta_1)$
    repeat
        $\triangleright$ Use gradient to update coefficients:
        $\beta_0 \leftarrow \beta_0 - \eta \cdot \frac{\partial}{\partial \beta_0} J(\beta_0, \beta_1)$ $\triangleright$ Take step in opposite direction of gradient
        $\beta_1 \leftarrow \beta_1 - \eta \cdot \frac{\partial}{\partial \beta_1} J(\beta_0, \beta_1)$
        oldCost $\leftarrow$ cost
        cost $\leftarrow J(\beta_0, \beta_1)$
    until $|\text{cost} - \text{oldCost}| < t$ $\triangleright$ Or some other "convergence" criterion
    Return β_0, β_1.
end procedure

Algorithm 2 Stochastic gradient descent for simple linear regression

procedure STOCHASTICGRADIENTDESCENT(dataset, step size η, tolerance t)
    Initialize β_0, β_1 arbitrarily.
    cost $\leftarrow J(\beta_0, \beta_1)$ ▷ This is computed using the dataset.
    repeat
        for $i = 1 \dots n$ **do**
            Choose a random data point (x_i, y_i) from dataset. ▷ The "stochastic" part.
            ▷ Use gradient of squared error on point (x_i, y_i) to update coefficients:
            $\beta_0 \leftarrow \beta_0 + \eta \cdot 2\,(y_i - h_\beta(x_i))$
            $\beta_1 \leftarrow \beta_1 + \eta \cdot 2\,(y_i - h_\beta(x_i))\,x_i$
        end for
        oldCost $\leftarrow$ cost
        cost $\leftarrow J(\beta_0, \beta_1)$
    until $|\text{cost} - \text{oldCost}| < t$
    Return β_0, β_1.
end procedure