Foundations of AI **Spring 2026**
Professor Serena Booth[1] **Machine Learning**

Linear Regression

An ice cream shop recorded its sales on various days of the year to try to determine if there is a relationship between temperature and sales. The data points are plotted below, in blue. The red line is an example of a simple linear regression obtained by minimizing the squared errors from the points to the line.

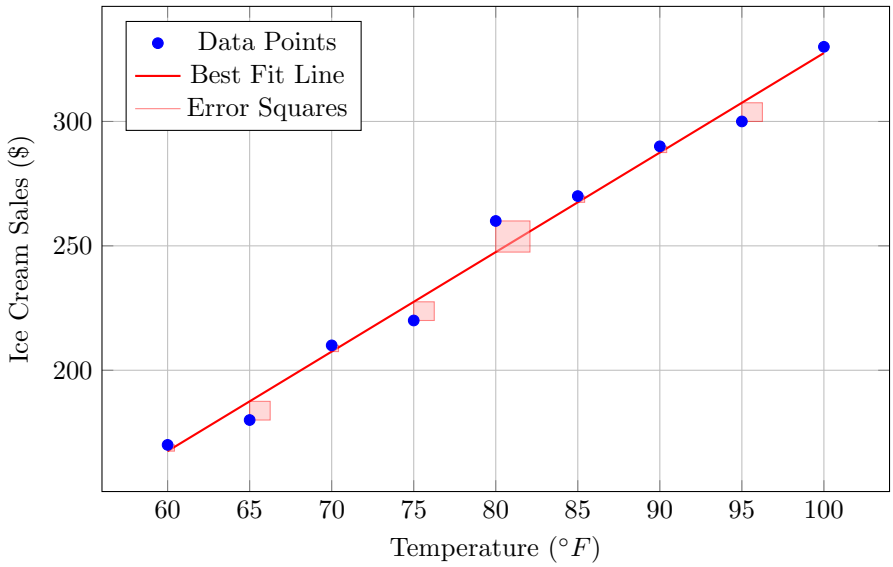

Figure 1: Temperature vs. Ice Cream Sales with Best Fit Line

1 Ordinary Least-Squares

Imagine we are given a data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \mid \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}, i \in \{1, \ldots, n\}\}$ of size n, where each data point $\boldsymbol{x}_i$ is d-dimensional, to which we hope to fit a linear model. When $d = 1$, a **linear model** is a model in which each y_i is approximated by $mx_i + b$. Here, as in high school, m is the slope, while b is the y intercept.

In a regression model, Y is called the response—or dependent—variable, while the features are called the explanatory variables, or the predictors, and we are said to be "regressing Y on X."

The feature values can be encoded in a matrix X:

$$X = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1d} \\ x_{21} & x_{22} & \ldots & x_{2d} \\ \vdots & & & \\ x_{n1} & x_{n2} & \ldots & x_{nd} \end{bmatrix}$$

Each $\boldsymbol{x}_i$ in X is a row vector, i.e., an element of $\mathbb{R}^d$. When the number of features $d = 1$, the problem is called **simple regression**. Otherwise, it is called **multiple regression**.

[1]This notes were compiled in conjunction with with Professors Eric Ewing and Amy Greenwald.

The response variables $\boldsymbol{y} \in \mathbb{R}^n$ are given by the column vector:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

The goal is to find a set of weights $\boldsymbol{w} \in \mathbb{R}^d$, or parameters, or **coefficients** (e.g., m and b), so that the **estimates**,[2] or **predictions**,[3] $X\boldsymbol{w}$ approximate $\boldsymbol{y}$:

$$X\boldsymbol{w} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} w_1 x_{11} + w_2 x_{12} + \ldots + w_d x_{1d} \\ w_2 x_{21} + w_2 x_{12} + \ldots + w_d x_{1d} \\ \vdots \\ w_1 x_{n1} + w_2 x_{n2} + \ldots + w_d x_{nd} \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \boldsymbol{y}$$

Observe that $X\boldsymbol{w} = \boldsymbol{y}$ is a system of n equations with d unknowns. A system of equations can be solved when it has the same number of equations and unknowns, i.e., when $n = d$. If $n < d$, i.e., if the matrix is wider than it is tall, then the system is **underdetermined**; it has infinitely many solutions. If $n > d$, i.e., if the matrix is taller than it is wide, which is usually the case in regression, the system is **overdetermined**, which means it has no solution. Consequently, rather than aim to solve this system of equations exactly, our goal is to solve for weights that minimize the difference between the predicted values $X\boldsymbol{w}$ and the observed values y. As usual, we square the errors, so as to make the ensuing optimization problem smooth.

Define the **residual** of a candidate solution $\boldsymbol{w}$ as the difference between $\boldsymbol{y}$ and $X\boldsymbol{w}$, i.e.,

$$\text{residual}(\boldsymbol{w}) = \boldsymbol{y} - X\boldsymbol{w}$$

The **least squares objective** is to minimize the square of this residual value: $\text{loss}(\boldsymbol{w}) \doteq \text{residual}^2(\boldsymbol{w})$.

As $\text{residual}(\boldsymbol{w})$ is an n-dimensional column vector, we square it as follows:

$$\text{loss}(\boldsymbol{w}) = \text{residual}^2(\boldsymbol{w}) = (\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w}) = \sum_{i=1}^{n} (\boldsymbol{y} - X\boldsymbol{w})_i^2$$

In other words, we are interested in minimizing the sum of the squared residuals.

As usual, to minimize an objective, we take its derivative and set it equal to zero.

First, let's simplify the loss:

$$\begin{aligned} &\text{loss}(\boldsymbol{w}) && (1) \\ &= (\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w}) && (2) \\ &= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T X\boldsymbol{w} - (X\boldsymbol{w})^T\boldsymbol{y} + (X\boldsymbol{w})^T X\boldsymbol{w} && (3) \\ &= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T X\boldsymbol{w} - \boldsymbol{w}^T X^T\boldsymbol{y} + \boldsymbol{w}^T X^T X\boldsymbol{w} && (4) \\ &= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{w}^T X^T\boldsymbol{y} + \boldsymbol{w}^T X^T X\boldsymbol{w} && (5) \end{aligned}$$

In Equation 3, we rely on the following fact: $(AB)^T = B^T A^T$. Equation 5 follows from the fact that $(\boldsymbol{y}^T X\boldsymbol{w})^T = \boldsymbol{w}^T X^T\boldsymbol{y}$, and that these values are both scalars, so they are equal to their transpose.

[2] statistical nomenclature

[3] machine learning nomenclature

Now let's take the derivative:

$$\nabla_{\boldsymbol{w}} \text{loss}(\boldsymbol{w}) \tag{6}$$
$$= \nabla_{\boldsymbol{w}} \boldsymbol{y}^T \boldsymbol{y} - 2\nabla_{\boldsymbol{w}} \boldsymbol{w}^T X^T \boldsymbol{y} + \nabla_{\boldsymbol{w}} \boldsymbol{w}^T X^T X \boldsymbol{w} \tag{7}$$
$$= -2X^T y + 2X^T X w \tag{8}$$

Equation 8 relies on the following fact: $\nabla_{\boldsymbol{x}} \boldsymbol{x}^T A \boldsymbol{x} = 2A\boldsymbol{x}$.

Setting this derivative equal to zero yields a **closed-form** (i.e., analytical) solution to linear regression:

$$-2X^T y + 2X^T X w = 0 \tag{9}$$
$$X^T X w = X^T y \tag{10}$$
$$w = \underbrace{(X^T X)^{-1} X^T}_{\text{pseudo-inverse}} y \tag{11}$$

The matrix $X^T X$ might not be invertible. This matrix is only invertible when the columns of X are linearly independent, so that they span $\mathbb{R}^d$. The pseudo-inverse $(X^T X)^{-1} X^T$, however, always exists, and can be calculated via a **singular value decomposition**.

We have established that $\boldsymbol{w}$ satisfies the **first-order optimality condition**, namely $\nabla_{\boldsymbol{w}} \text{loss}(\boldsymbol{w}) = 0$. To conclude that $\boldsymbol{w}$ is a minimum, we also need to establish that it satisfies the second-order optimality condition. For $\boldsymbol{w}$ to be a minimum, the **second-order optimality condition** requires that $\nabla^2_{\boldsymbol{w}} f(\boldsymbol{w}) \geq 0$. Indeed, $\nabla^2_{\boldsymbol{w}} \text{loss}(\boldsymbol{w}) = 2X^T X \geq 0$.

Alternative Derivation We describe an alternative shorter, geometric, and arguably more intuitive way, to derive the least-squares estimators. The only way the residual $\boldsymbol{y} - X\boldsymbol{w}$ can be zero is if $\boldsymbol{y}$ is a linear combination of the columns of X: i.e., if $\boldsymbol{y}$ lies in the column span of X. As most of the time it does not, our stated goal of minimizing the value of the residual (squared) can instead be understood as minimizing the distance from the residual to the column span of X. To minimize this distance, the residual should be **projected** onto the column span of X; that is, it should be **orthogonal** to each column of X: for all $j \in \{1, \ldots, n\}$, i.e., $(\boldsymbol{y} - X\boldsymbol{w}) \cdot X_j = 0$. Equivalently, but expressed more compactly in matrix notation, $(\boldsymbol{y} - X\boldsymbol{w})^T X = 0$. This requirement implies:

$$X^T(\boldsymbol{y} - X\boldsymbol{w}) = 0 \tag{12}$$
$$X^T X \boldsymbol{w} = X^T \boldsymbol{y} \tag{13}$$
$$\boldsymbol{w} = (X^T X)^{-1} X^T \boldsymbol{y} \tag{14}$$