Linear regression

COMS 4721 Spring 2022
Daniel Hsu

Motivation for Linear Models

Motivation

Linear model: Statistical model (for supervised learning) based on linear structure

- Linearity (a.k.a. superposition) is a well-understood and powerful property
- Linearity enables particular type of extrapolation
- Can be readily upgraded to increase expressiveness
- At the heart of many statistical and ML methods

Ordinary Least Squares

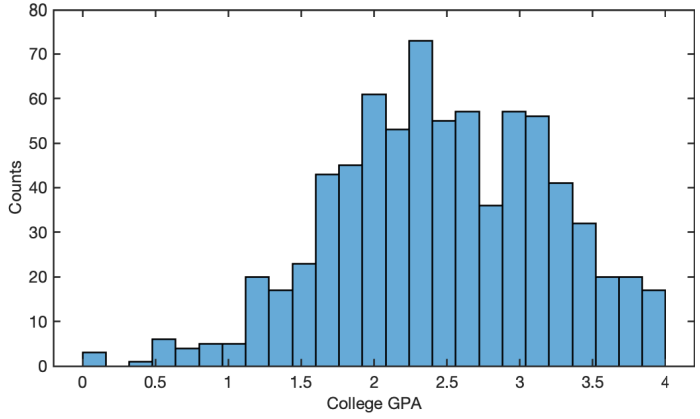
Example #1: Predicting college GPA

Task: Predict a student's final college grade point average (GPA) even before they start

Example #1: Predicting college GPA

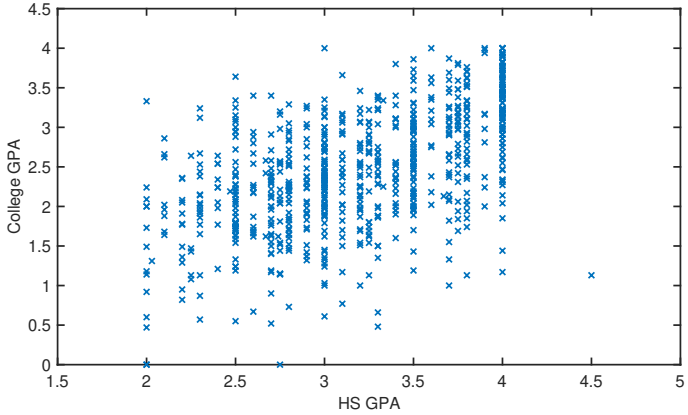
Task: Predict a student's final college grade point average (GPA) even before they start

- Data: 750 Dartmouth students' `College GPA`
- Mean: 2.46; standard deviation: 0.746



Example #1: Predicting college GPA from high school GPA

- Also available at time of prediction: student's high school GPA (`HS GPA`)



Least squares linear regression

Task: Given collection $\mathcal{S}$ of n labeled examples $(x, y) \in \mathbb{R} \times \mathbb{R}$, find function f of the form

$$f(x) = mx + \theta$$

that minimizes the **sum of squared errors (SSE)**

$$\text{sse}(m, \theta; \mathcal{S}) := \sum_{(x,y)\in\mathcal{S}} (mx + \theta - y)^2$$

(If divide by $n = |\mathcal{S}|$, we get **mean squared error (MSE)**)

Least squares linear regression

Task: Given collection $\mathcal{S}$ of n labeled examples $(x, y) \in \mathbb{R} \times \mathbb{R}$, find function f of the form

$$f(x) = mx + \theta$$

that minimizes the **sum of squared errors (SSE)**

$$\text{sse}(m, \theta; \mathcal{S}) := \sum_{(x,y)\in\mathcal{S}} (mx + \theta - y)^2$$

(If divide by $n = |\mathcal{S}|$, we get **mean squared error (MSE)**)

- m is **slope**; θ is **intercept** (a.k.a. **y-intercept**)
- **Technicality:** f described above is an **affine** function; it is **linear** only if $\theta = 0$
  - Distinction between affine & linear usually unimportant in this context (and often ignored)

Least squares linear regression

Task: Given collection $\mathcal{S}$ of n labeled examples $(x, y) \in \mathbb{R} \times \mathbb{R}$, find function f of the form

$$f(x) = mx + \theta$$

that minimizes the **sum of squared errors (SSE)**

$$\text{sse}(m, \theta; \mathcal{S}) := \sum_{(x,y)\in\mathcal{S}} (mx + \theta - y)^2$$

(If divide by $n = |\mathcal{S}|$, we get **mean squared error (MSE)**)

- m is **slope**; θ is **intercept** (a.k.a. **y-intercept**)
- **Technicality:** f described above is an **affine** function; it is **linear** only if $\theta = 0$
  - Distinction between affine & linear usually unimportant in this context (and often ignored)

College GPA prediction example:

- Affine function with positive slope means higher `HS GPA` gets higher prediction of `College GPA`
- Does a linear/affine relationship make sense?

Ordinary least squares

Ordinary least squares (OLS): Given labeled examples $\mathcal{S}$ from $\mathbb{R} \times \mathbb{R}$, return (m, θ) minimizing

$$\text{sse}(m, \theta; \mathcal{S}) = \sum_{(x,y) \in \mathcal{S}} (mx + \theta - y)^2$$

Ordinary least squares

Ordinary least squares (OLS): Given labeled examples $\mathcal{S}$ from $\mathbb{R} \times \mathbb{R}$, return (m, θ) minimizing

$$\text{sse}(m, \theta; \mathcal{S}) = \sum_{(x,y) \in \mathcal{S}} (mx + \theta - y)^2$$

- OLS is defined as solution to a mathematical optimization problem
- To derive an algorithm that finds a solution:
  1. Use calculus to characterize solution
  2. Find an algorithm that produces a solution with the desired characteristics

Ordinary least squares

Ordinary least squares (OLS): Given labeled examples $\mathcal{S}$ from $\mathbb{R} \times \mathbb{R}$, return (m, θ) minimizing

$$\text{sse}(m, \theta; \mathcal{S}) = \sum_{(x,y) \in \mathcal{S}} (mx + \theta - y)^2$$

- OLS is defined as solution to a mathematical optimization problem
- To derive an algorithm that finds a solution:
  1. Use calculus to characterize solution
  2. Find an algorithm that produces a solution with the desired characteristics

Adrien-Marie Legendre, 1805

Carl Friedrich Gauss, 1809

Using calculus to characterize the solution

Theory of calculus tells us that minimizer (m, θ) of $\mathrm{sse}(m, \theta; \mathcal{S})$ must satisfy

$$\frac{\partial}{\partial m} \left\{ \mathrm{sse}(m, \theta; \mathcal{S}) \right\} = 0$$

$$\frac{\partial}{\partial \theta} \left\{ \mathrm{sse}(m, \theta; \mathcal{S}) \right\} = 0$$

Using calculus to characterize the solution

Theory of calculus tells us that minimizer (m, θ) of $\mathrm{sse}(m, \theta; \mathcal{S})$ must satisfy

$$\frac{\partial}{\partial m} \left\{ \sum_{(x,y) \in \mathcal{S}} (mx + \theta - y)^2 \right\} = 0$$

$$\frac{\partial}{\partial \theta} \left\{ \sum_{(x,y) \in \mathcal{S}} (mx + \theta - y)^2 \right\} = 0$$

Using calculus to characterize the solution

Theory of calculus tells us that minimizer (m, θ) of $\mathrm{sse}(m, \theta; \mathcal{S})$ must satisfy

$$\sum_{(x,y)\in\mathcal{S}} x(mx + \theta - y) = 0$$

$$\sum_{(x,y)\in\mathcal{S}} (mx + \theta - y) = 0$$

These linear equations, in the two unknowns (m, θ), are called the **normal equations**

Using calculus to characterize the solution

Theory of calculus tells us that minimizer (m, θ) of $\text{sse}(m, \theta; \mathcal{S})$ must satisfy

$$\sum_{(x,y)\in\mathcal{S}} x(mx + \theta - y) = 0$$

$$\sum_{(x,y)\in\mathcal{S}} (mx + \theta - y) = 0$$

These linear equations, in the two unknowns (m, θ), are called the **normal equations**

In fact, satisfying normal equations is also sufficient to minimize SSE!

Theorem. If $(m_{\text{ols}}, \theta_{\text{ols}})$ satisfy the normal equations for $\mathcal{S}$, then

$$\text{sse}(m_{\text{ols}}, \theta_{\text{ols}}; \mathcal{S}) = \min_{(m,\theta)\in\mathbb{R}\times\mathbb{R}} \text{sse}(m, \theta; \mathcal{S})$$

Equivalent ways to write the normal equations

Some notation simplifies the equations:

$$\overline{x} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} x \qquad \overline{x^2} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} x^2$$

$$\overline{xy} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} xy \qquad \overline{y} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} y$$

- Normal equations using above notation:

$$\begin{aligned} \overline{x^2} \cdot m &+ \overline{x} \cdot \theta &&= \overline{xy} \\ \overline{x} \cdot m &+ \theta &&= \overline{y} \end{aligned}$$

Equivalent ways to write the normal equations

Some notation simplifies the equations:

$$\overline{x} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} x \qquad \overline{x^2} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} x^2$$

$$\overline{xy} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} xy \qquad \overline{y} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} y$$

- Normal equations using above notation:

$$\begin{aligned} \overline{x^2} \cdot m &+ \overline{x} \cdot \theta &&= \overline{xy} \\ \overline{x} \cdot m &+ \theta &&= \overline{y} \end{aligned}$$

- Normal equations in matrix form:

$$\begin{bmatrix} \overline{x^2} & \overline{x} \\ \overline{x} & 1 \end{bmatrix} \begin{bmatrix} m \\ \theta \end{bmatrix} = \begin{bmatrix} \overline{xy} \\ \overline{y} \end{bmatrix}$$

Equivalent ways to write the normal equations

Some notation simplifies the equations:

$$\overline{x} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} x \qquad \overline{x^2} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} x^2$$

$$\overline{xy} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} xy \qquad \overline{y} := \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} y$$

- Normal equations using above notation:

$$\begin{aligned} \overline{x^2} \cdot m &+ \overline{x} \cdot \theta &&= \overline{xy} \\ \overline{x} \cdot m &+ \theta &&= \overline{y} \end{aligned}$$

- Normal equations in matrix form:

$$\begin{bmatrix} \overline{x^2} & \overline{x} \\ \overline{x} & 1 \end{bmatrix} \begin{bmatrix} m \\ \theta \end{bmatrix} = \begin{bmatrix} \overline{xy} \\ \overline{y} \end{bmatrix}$$

Solution via elementary algebra:

$$m = \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2}, \quad \theta = \overline{y} - \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2} \cdot \overline{x}$$

Solving the normal equations

Formula for solution to normal equations

$$m = \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2}, \quad \theta = \overline{y} - \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2} \cdot \overline{x}$$

only makes sense if fractions have non-zero denominators

Solving the normal equations

Formula for solution to normal equations

$$m = \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2}, \quad \theta = \overline{y} - \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2} \cdot \overline{x}$$

only makes sense if fractions have non-zero denominators

If $\overline{x^2} - \overline{x}^2 = 0$, system of equations has only one linearly independent equation

$$\overline{x} \cdot m + \theta = \overline{y}$$

which is always satisfied by

$$m = 0, \quad \theta = \overline{y}$$

Solving the normal equations

Formula for solution to normal equations

$$m = \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2}, \quad \theta = \overline{y} - \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2} \cdot \overline{x}$$

only makes sense if fractions have non-zero denominators

If $\overline{x^2} - \overline{x}^2 = 0$, system of equations has only one linearly independent equation

$$\overline{x} \cdot m + \theta = \overline{y}$$

which is always satisfied by

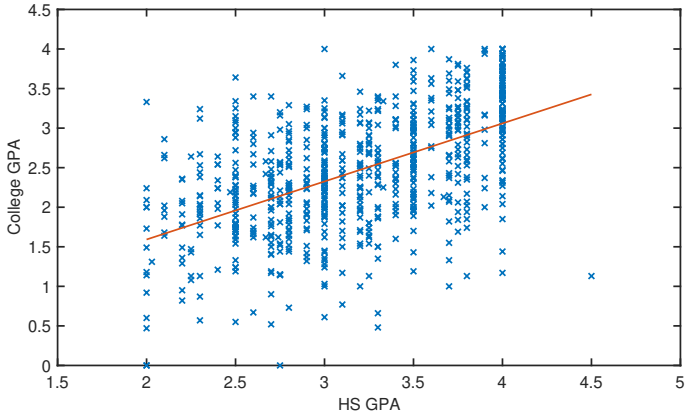
$$m = 0, \quad \theta = \overline{y}$$

OLS (simple implementation): Given labeled examples $\mathcal{S}$ from $\mathbb{R} \times \mathbb{R}$,

- If $\overline{x^2} - \overline{x}^2 \neq 0$, return $m := \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2}$ and $\theta := \overline{y} - \frac{\overline{xy} - \overline{x} \cdot \overline{y}}{\overline{x^2} - \overline{x}^2} \cdot \overline{x}$
- If $\overline{x^2} - \overline{x}^2 = 0$, return $m := 0$ and $\theta := \overline{y}$

Example #1: Using OLS

Plotting $f(x) = m_{\text{ols}}x + \theta_{\text{ols}}$, with $(m_{\text{ols}}, \theta_{\text{ols}}) \approx (0.7337, 0.1247)$ obtained from OLS:



Interpretation: Unit increase in `HS GPA` $\Rightarrow$ increase of m_{ols} in prediction of `College GPA`

Example #1: Root mean squared error

Root mean squared error (RMSE): square-root of MSE

Example #1: Root mean squared error

Root mean squared error (RMSE): square-root of MSE

RMSE of $(m_{\text{ols}}, \theta_{\text{ols}})$ from OLS:

$$\sqrt{\frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{S}} (m_{\text{ols}} x + \theta_{\text{ols}} - y)^2} = 0.6248$$

Standard deviation of `College GPA` in $\mathcal{S}$

$$\sqrt{\frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{S}} (\overline{y} - y)^2} = 0.746$$

Example #1: Root mean squared error

Root mean squared error (RMSE): square-root of MSE

RMSE of $(m_{\text{ols}}, \theta_{\text{ols}})$ from OLS:

$$\sqrt{\frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{S}} (m_{\text{ols}} x + \theta_{\text{ols}} - y)^2} = 0.6248$$

Standard deviation of `College GPA` in $\mathcal{S}$

$$\sqrt{\frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{S}} (\overline{y} - y)^2} = 0.746$$

But both $\overline{y}$ and $(m_{\text{ols}}, \theta_{\text{ols}})$ depend on $\mathcal{S}$!

Example #1: Root mean squared error

Root mean squared error (RMSE): square-root of MSE

RMSE of $(m_{\text{ols}}, \theta_{\text{ols}})$ from OLS:

$$\sqrt{\frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{S}} (m_{\text{ols}} x + \theta_{\text{ols}} - y)^2} = 0.6248$$

Standard deviation of `College GPA` in $\mathcal{S}$

$$\sqrt{\frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{S}} (\overline{y} - y)^2} = 0.746$$

But both $\overline{y}$ and $(m_{\text{ols}}, \theta_{\text{ols}})$ depend on $\mathcal{S}$! Fortunately, we have 250 test examples not included in $\mathcal{S}$.

Test RMSE of $(m_{\text{ols}}, \theta_{\text{ols}})$:	0.6120
Test RMSE of $\overline{y}$:	0.7237

Example #1: More features

In addition to `HS GPA`, also have `SAT Score` for every student at time of prediction.

Example #1: More features

In addition to `HS GPA`, also have `SAT Score` for every student at time of prediction.

New task: Given collection $\mathcal{S}$ of n labeled examples $(\vec{x}, y) \in \mathbb{R}^2 \times \mathbb{R}$, find function f of the form

$$f(\vec{x}) = m_1 x_1 + m_2 x_2 + \theta$$

that minimizes SSE

$$\text{sse}(m_1, m_2, \theta; \mathcal{S}) := \sum_{(\vec{x}, y) \in \mathcal{S}} (m_1 x_1 + m_2 x_2 + \theta - y)^2$$

Example #1: More features

In addition to `HS GPA`, also have `SAT Score` for every student at time of prediction.

New task: Given collection $\mathcal{S}$ of n labeled examples $(\vec{x}, y) \in \mathbb{R}^2 \times \mathbb{R}$, find function f of the form

$$f(\vec{x}) = m_1 x_1 + m_2 x_2 + \theta$$

that minimizes SSE

$$\text{sse}(m_1, m_2, \theta; \mathcal{S}) := \sum_{(\vec{x}, y) \in \mathcal{S}} (m_1 x_1 + m_2 x_2 + \theta - y)^2$$

Normal equations: system of three linear equations in three unknowns (m_1, m_2, θ) ...

Example #1: More features

In addition to `HS GPA`, also have `SAT Score` for every student at time of prediction.

New task: Given collection $\mathcal{S}$ of n labeled examples $(\vec{x}, y) \in \mathbb{R}^2 \times \mathbb{R}$, find function f of the form

$$f(\vec{x}) = m_1 x_1 + m_2 x_2 + \theta$$

that minimizes SSE

$$\text{sse}(m_1, m_2, \theta; \mathcal{S}) := \sum_{(\vec{x}, y) \in \mathcal{S}} (m_1 x_1 + m_2 x_2 + \theta - y)^2$$

Normal equations: system of three linear equations in three unknowns (m_1, m_2, θ) ...

Found $(m_{\text{ols},1}, m_{\text{ols},2}, \theta_{\text{ols}}) \approx (0.0016, 0.5585, -0.9405)$

Training RMSE of $(m_{\text{ols},1}, m_{\text{ols},2}, \theta_{\text{ols}})$:	0.5904
Test RMSE of $(m_{\text{ols},1}, m_{\text{ols},2}, \theta_{\text{ols}})$:	0.6030

Consequences of linearity

(Recall $(m_{\text{ols},1}, m_{\text{ols},2}, \theta_{\text{ols}}) \approx (0.0016, 0.5585, -0.9405)$)

Consider affine function $f\colon \mathbb{R}^2 \to \mathbb{R}$ given by

$$f(\vec{x}) = m_{\text{ols},1} x_1 + m_{\text{ols},2} x_2 + \theta_{\text{ols}}$$

($\vec{x} = (x_1, x_2)$ where $x_1 =$ `SAT Score` and $x_2 =$ `HS GPA`; $f(\vec{x}) =$ predicted `College GPA`)

Consequences of linearity

(Recall $(m_{\text{ols},1}, m_{\text{ols},2}, \theta_{\text{ols}}) \approx (0.0016, 0.5585, -0.9405)$)

Consider affine function $f\colon \mathbb{R}^2 \to \mathbb{R}$ given by

$$f(\vec{x}) = m_{\text{ols},1} x_1 + m_{\text{ols},2} x_2 + \theta_{\text{ols}}$$

($\vec{x} = (x_1, x_2)$ where $x_1 =$ `SAT Score` and $x_2 =$ `HS GPA`; $f(\vec{x}) =$ predicted `College GPA`)

- What is predicted `College GPA` of student with `SAT Score` $= 0$ and `HS GPA` $= 0$?

Consequences of linearity

(Recall $(m_{\text{ols},1}, m_{\text{ols},2}, \theta_{\text{ols}}) \approx (0.0016, 0.5585, -0.9405)$)

Consider affine function $f\colon \mathbb{R}^2 \to \mathbb{R}$ given by

$$f(\vec{x}) = m_{\text{ols},1} x_1 + m_{\text{ols},2} x_2 + \theta_{\text{ols}}$$

($\vec{x} = (x_1, x_2)$ where $x_1 =$ `SAT Score` and $x_2 =$ `HS GPA`; $f(\vec{x}) =$ predicted `College GPA`)

- What is predicted `College GPA` of student with `SAT Score` $= 0$ and `HS GPA` $= 0$?
- Suppose f predicts Alice's `College GPA` is 3.5, and f predicts Bob's `College GPA` is 4.5.
  - What does f predict for a student whose `SAT Score` and `HS GPA` are, respectively, exactly midway between of those of Alice and Bob?

Consequences of linearity

(Recall $(m_{\text{ols},1}, m_{\text{ols},2}, \theta_{\text{ols}}) \approx (0.0016, 0.5585, -0.9405)$)

Consider affine function $f\colon \mathbb{R}^2 \to \mathbb{R}$ given by

$$f(\vec{x}) = m_{\text{ols},1} x_1 + m_{\text{ols},2} x_2 + \theta_{\text{ols}}$$

($\vec{x} = (x_1, x_2)$ where $x_1 =$ `SAT Score` and $x_2 =$ `HS GPA`; $f(\vec{x}) =$ predicted `College GPA`)

- What is predicted `College GPA` of student with `SAT Score` $= 0$ and `HS GPA` $= 0$?
- Suppose f predicts Alice's `College GPA` is 3.5, and f predicts Bob's `College GPA` is 4.5.
  - What does f predict for a student whose `SAT Score` and `HS GPA` are, respectively, exactly midway between of those of Alice and Bob?
  - Suppose you also know that Bob's `SAT Score` is the same as that of Alice, but that Bob's `HS GPA` is $1.5\times$ that of Alice. Can you determine Alice's `SAT Score` and `HS GPA`?

Consequences of linearity

(Recall $(m_{\text{ols},1}, m_{\text{ols},2}, \theta_{\text{ols}}) \approx (0.0016, 0.5585, -0.9405)$)

Consider affine function $f\colon \mathbb{R}^2 \to \mathbb{R}$ given by

$$f(\vec{x}) = m_{\text{ols},1} x_1 + m_{\text{ols},2} x_2 + \theta_{\text{ols}}$$

($\vec{x} = (x_1, x_2)$ where $x_1 =$ `SAT Score` and $x_2 =$ `HS GPA`; $f(\vec{x}) =$ predicted `College GPA`)

- What is predicted `College GPA` of student with `SAT Score` $= 0$ and `HS GPA` $= 0$?
- Suppose f predicts Alice's `College GPA` is 3.5, and f predicts Bob's `College GPA` is 4.5.
  - What does f predict for a student whose `SAT Score` and `HS GPA` are, respectively, exactly midway between of those of Alice and Bob?
  - Suppose you also know that Bob's `SAT Score` is the same as that of Alice, but that Bob's `HS GPA` is $1.5\times$ that of Alice. Can you determine Alice's `SAT Score` and `HS GPA`?
  - What if, instead, Bob's `SAT Score` and `HS GPA` are both $1.5\times$ those of Alice—can you determine Alice's `SAT Score` and `HS GPA` under this supposition?

Multivariate OLS

Multivariate least squares linear regression

Task: Given collection $\mathcal{S}$ of n labeled examples $(\vec{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, find linear function f of the form

$$f(\vec{x}) = \vec{x} \cdot \vec{w} = w_1 x_1 + \cdots + w_d x_d$$

that minimizes SSE

$$\text{sse}(\vec{w}; \mathcal{S}) := \sum_{(\vec{x}, y) \in \mathcal{S}} (\vec{x} \cdot \vec{w} - y)^2$$

Multivariate least squares linear regression

Task: Given collection $\mathcal{S}$ of n labeled examples $(\vec{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, find linear function f of the form

$$f(\vec{x}) = \vec{x} \cdot \vec{w} = w_1 x_1 + \cdots + w_d x_d$$

that minimizes SSE

$$\text{sse}(\vec{w}; \mathcal{S}) := \sum_{(\vec{x}, y) \in \mathcal{S}} (\vec{x} \cdot \vec{w} - y)^2$$

- Vector $\vec{w} \in \mathbb{R}^d$ is called a **weight vector** (a.k.a. **coefficient vector**)

Multivariate least squares linear regression

Task: Given collection $\mathcal{S}$ of n labeled examples $(\vec{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, find linear function f of the form

$$f(\vec{x}) = \vec{x} \cdot \vec{w} = w_1 x_1 + \cdots + w_d x_d$$

that minimizes SSE

$$\text{sse}(\vec{w}; \mathcal{S}) := \sum_{(\vec{x}, y) \in \mathcal{S}} (\vec{x} \cdot \vec{w} - y)^2$$

- Vector $\vec{w} \in \mathbb{R}^d$ is called a **weight vector** (a.k.a. **coefficient vector**)
- **Question:** What happened to the intercept θ?

Multivariate least squares linear regression

Task: Given collection $\mathcal{S}$ of n labeled examples $(\vec{x}, y) \in \mathbb{R}^d \times \mathbb{R}$, find linear function f of the form

$$f(\vec{x}) = \vec{x} \cdot \vec{w} = w_1 x_1 + \cdots + w_d x_d$$

that minimizes SSE

$$\text{sse}(\vec{w}; \mathcal{S}) := \sum_{(\vec{x}, y) \in \mathcal{S}} (\vec{x} \cdot \vec{w} - y)^2$$

- Vector $\vec{w} \in \mathbb{R}^d$ is called a **weight vector** (a.k.a. **coefficient vector**)
- **Question:** What happened to the intercept θ?
- **Answer:** If you want it, work in dimension $d + 1$:
  - Replace feature vector $\vec{x} = (x_1, \ldots, x_d)$ with

    $$\vec{x}' := (x_1, \ldots, x_d, 1) \in \mathbb{R}^{d+1}$$

  - For $\vec{w} = (w_1, \ldots, w_d, w_{d+1})$,

    $$\vec{x}' \cdot \vec{w} = w_1 x_1 + \cdots + w_d x_d + w_{d+1}$$

    so w_{d+1} plays the role of the intercept

Using multivariable calculus to characterize the solution

Theory of calculus tells us that minimizer $\vec{w}$ of $\mathrm{sse}(\vec{w}; \mathcal{S})$ must satisfy

$$\frac{\partial}{\partial w_1} \left\{ \sum_{(\vec{x},y)\in\mathcal{S}} (\vec{x}\cdot\vec{w} - y)^2 \right\} = 0$$

$$\vdots$$

$$\frac{\partial}{\partial w_d} \left\{ \sum_{(\vec{x},y)\in\mathcal{S}} (\vec{x}\cdot\vec{w} - y)^2 \right\} = 0$$

Using multivariable calculus to characterize the solution

Theory of calculus tells us that minimizer $\vec{w}$ of $\mathrm{sse}(\vec{w}; \mathcal{S})$ must satisfy

$$\sum_{(\vec{x},y)\in\mathcal{S}} x_1 \ (\vec{x}\cdot\vec{w} - y) \ = 0$$

$$\vdots$$

$$\sum_{(\vec{x},y)\in\mathcal{S}} x_d \ (\vec{x}\cdot\vec{w} - y) \ = 0$$

Using multivariable calculus to characterize the solution

Theory of calculus tells us that minimizer $\vec{w}$ of $\text{sse}(\vec{w}; \mathcal{S})$ must satisfy

$$\begin{aligned} \sum_{(\vec{x},y)\in\mathcal{S}} x_1 \ (\vec{x}\cdot\vec{w} - y) \ &= 0 \\ &\vdots \\ \sum_{(\vec{x},y)\in\mathcal{S}} x_d \ (\vec{x}\cdot\vec{w} - y) \ &= 0 \end{aligned}$$

Normal equations: the system of d linear equations in d unknowns $(w_1, \ldots, w_d)$ shown above

Theorem. If $\vec{w}_{\text{ols}}$ satisfies the normal equations for $\mathcal{S}$, then

$$\text{sse}(\vec{w}_{\text{ols}}; \mathcal{S}) = \min_{\vec{w}\in\mathbb{R}^d} \text{sse}(\vec{w}; \mathcal{S})$$

Multivariate OLS

Many efficient algorithms for finding solution to system of linear equations (e.g., Gaussian elimination)

OLS (multivariate): Given labeled examples $\mathcal{S}$ from $\mathbb{R}^d \times \mathbb{R}$, return solution $\vec{w}$ to normal equations

$$\sum_{(\vec{x},y)\in\mathcal{S}} x_1 \ (\vec{x}\cdot\vec{w} - y) = 0$$

$$\vdots$$

$$\sum_{(\vec{x},y)\in\mathcal{S}} x_d \ (\vec{x}\cdot\vec{w} - y) = 0$$

Example #2: Predicting yields of mesquite bushes

(Adapted from Gelman and Hill, 2007)

Task: Predict total "biomass" post-harvest, using visual measurements of a mesquite bush pre-harvest

Example #2: Predicting yields of mesquite bushes

(Adapted from Gelman and Hill, 2007)

Task: Predict total "biomass" post-harvest, using visual measurements of a mesquite bush pre-harvest

- Data: measurements and labels for $n = 26$ bushes
  - `Diam1`: diameter of the canopy in meters, measured along the longer axis of the bush
  - `Diam2`: diameter of the canopy in meters, measured along the shorter axis
  - `TotHt`: total height of the bush in meters
  - `CanHt`: height of the canopy in meters
  - `Dens`: plant unit density (i.e., number of primary stems per plant unit)
  - `LogLeafWt`: $\log_{10}$[weight of photosynthetic material in grams, post-harvest]
- Sample mean of `LogLeafWt`: 2.59; sample stddev of `LogLeafWt`: 0.44

Example #2: Predicting yields of mesquite bushes

(Adapted from Gelman and Hill, 2007)

Task: Predict total "biomass" post-harvest, using visual measurements of a mesquite bush pre-harvest

- Data: measurements and labels for $n = 26$ bushes
  - `Diam1`: diameter of the canopy in meters, measured along the longer axis of the bush
  - `Diam2`: diameter of the canopy in meters, measured along the shorter axis
  - `TotHt`: total height of the bush in meters
  - `CanHt`: height of the canopy in meters
  - `Dens`: plant unit density (i.e., number of primary stems per plant unit)
  - `LogLeafWt`: $\log_{10}$[weight of photosynthetic material in grams, post-harvest]
- Sample mean of `LogLeafWt`: 2.59; sample stddev of `LogLeafWt`: 0.44

Diam1	Diam2	TotHt	CanHt	Dens	LogLeafWt
2.80	1.70	1.70	1.20	1	2.58
1.30	0.95	1.35	0.95	1	2.43
2.40	1.30	1.50	0.90	2	2.51
2.40	2.40	1.60	1.10	3	2.84
⋮	⋮	⋮	⋮	⋮	⋮

(One training example per row)

Example #2: An affine function via (multivariate) OLS

Want: Affine function of (`Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`) that minimizes SSE in predicting `LogLeafWt` over the 26 training examples

Example #2: An affine function via (multivariate) OLS

Want: Affine function of (`Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`) that minimizes SSE in predicting `LogLeafWt` over the 26 training examples

"data matrix" with constant feature appended to each feature vector

		`Diam1`	`Diam2`	`TotHt`	`CanHt`	`Dens`	`const`			`LogLeafWt`
$\vec{x}_1 =$	(	2.80	1.70	1.70	1.20	1	1	),	$y_1 =$	2.58
$\vec{x}_2 =$	(	1.30	0.95	1.35	0.95	1	1	),	$y_2 =$	2.43
$\vec{x}_3 =$	(	2.40	1.30	1.50	0.90	2	1	),	$y_3 =$	2.51
$\vec{x}_4 =$	(	2.40	2.40	1.60	1.10	3	1	),	$y_4 =$	2.84
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$

Example #2: An affine function via (multivariate) OLS

Want: Affine function of (`Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`) that minimizes SSE in predicting `LogLeafWt` over the 26 training examples

"data matrix" with `const` feature appended to each feature vector

		`Diam1`	`Diam2`	`TotHt`	`CanHt`	`Dens`	`const`			`LogLeafWt`
$\vec{x}_1 =$	(	2.80	1.70	1.70	1.20	1	1	),	$y_1 =$	2.58
$\vec{x}_2 =$	(	1.30	0.95	1.35	0.95	1	1	),	$y_2 =$	2.43
$\vec{x}_3 =$	(	2.40	1.30	1.50	0.90	2	1	),	$y_3 =$	2.51
$\vec{x}_4 =$	(	2.40	2.40	1.60	1.10	3	1	),	$y_4 =$	2.84
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$

Form the normal equations in 6 variables $(w_1, w_2, w_3, w_4, w_5, w_6)$:

Example #2: An affine function via (multivariate) OLS

Want: Affine function of (`Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`) that minimizes SSE in predicting `LogLeafWt` over the 26 training examples

"data matrix" with constant feature appended to each feature vector

		`Diam1`	`Diam2`	`TotHt`	`CanHt`	`Dens`	`const`			`LogLeafWt`
$\vec{x}_1 =$	(	2.80	1.70	1.70	1.20	1	1	),	$y_1 =$	2.58
$\vec{x}_2 =$	(	1.30	0.95	1.35	0.95	1	1	),	$y_2 =$	2.43
$\vec{x}_3 =$	(	2.40	1.30	1.50	0.90	2	1	),	$y_3 =$	2.51
$\vec{x}_4 =$	(	2.40	2.40	1.60	1.10	3	1	),	$y_4 =$	2.84
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$

Form the normal equations in 6 variables $(w_1, w_2, w_3, w_4, w_5, w_6)$:

$$
\begin{array}{rcrcrcrcrcrcr}
58.600w_1 & + & 42.2800w_2 & + & 40.2500w_3 & + & 30.9500w_4 & + & 44.00w_5 & + & 26.00w_6 & = & 67.4288 \\
160.360w_1 & + & 117.7840w_2 & + & 101.0250w_3 & + & 78.2350w_4 & + & 128.50w_5 & + & 58.60w_6 & = & 162.4314 \\
117.784w_1 & + & 90.5344w_2 & + & 74.9195w_3 & + & 58.2755w_4 & + & 95.38w_5 & + & 42.28w_6 & = & 118.9825 \\
101.025w_1 & + & 74.9195w_2 & + & 68.2775w_3 & + & 52.7375w_4 & + & 81.20w_5 & + & 40.25w_6 & = & 109.0527 \\
78.235w_1 & + & 58.2755w_2 & + & 52.7375w_3 & + & 41.2675w_4 & + & 63.70w_5 & + & 30.95w_6 & = & 84.0794 \\
128.500w_1 & + & 95.3800w_2 & + & 81.2000w_3 & + & 63.7000w_4 & + & 140.00w_5 & + & 44.00w_6 & = & 124.4754
\end{array}
$$

Example #2: An affine function via (multivariate) OLS

Want: Affine function of (`Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`) that minimizes SSE in predicting `LogLeafWt` over the 26 training examples

"data matrix" with constant feature appended to each feature vector

		`Diam1`	`Diam2`	`TotHt`	`CanHt`	`Dens`	`const`			`LogLeafWt`
$\vec{x}_1 =$	(	2.80	1.70	1.70	1.20	1	1	),	$y_1 =$	2.58
$\vec{x}_2 =$	(	1.30	0.95	1.35	0.95	1	1	),	$y_2 =$	2.43
$\vec{x}_3 =$	(	2.40	1.30	1.50	0.90	2	1	),	$y_3 =$	2.51
$\vec{x}_4 =$	(	2.40	2.40	1.60	1.10	3	1	),	$y_4 =$	2.84
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$

Form the normal equations in 6 variables $(w_1, w_2, w_3, w_4, w_5, w_6)$:

$$
\begin{array}{rcrcrcrcrcrcr}
58.600w_1 & + & 42.2800w_2 & + & 40.2500w_3 & + & 30.9500w_4 & + & 44.00w_5 & + & 26.00w_6 & = & 67.4288 \\
160.360w_1 & + & 117.7840w_2 & + & 101.0250w_3 & + & 78.2350w_4 & + & 128.50w_5 & + & 58.60w_6 & = & 162.4314 \\
117.784w_1 & + & 90.5344w_2 & + & 74.9195w_3 & + & 58.2755w_4 & + & 95.38w_5 & + & 42.28w_6 & = & 118.9825 \\
101.025w_1 & + & 74.9195w_2 & + & 68.2775w_3 & + & 52.7375w_4 & + & 81.20w_5 & + & 40.25w_6 & = & 109.0527 \\
78.235w_1 & + & 58.2755w_2 & + & 52.7375w_3 & + & 41.2675w_4 & + & 63.70w_5 & + & 30.95w_6 & = & 84.0794 \\
128.500w_1 & + & 95.3800w_2 & + & 81.2000w_3 & + & 63.7000w_4 & + & 140.00w_5 & + & 44.00w_6 & = & 124.4754
\end{array}
$$

Solve! Solution is $(0.1627, 0.1552, 0.3414, -0.0008, -0.0391, 1.513)$

Example #2: Learned affine function

Solution to normal equations: $(0.1627, 0.1552, 0.3414, -0.0008, -0.0391, 1.513)$

Example #2: Learned affine function

Solution to normal equations: $(0.1627, 0.1552, 0.3414, -0.0008, -0.0391, 1.513)$

Learned affine function:

$$f_{\text{ols}}(\texttt{Diam1},\texttt{Diam2},\texttt{TotHt},\texttt{CanHt},\texttt{Dens}) =$$
$$0.1627 \times \texttt{Diam1} + 0.1552 \times \texttt{Diam2} + 0.3414 \times \texttt{TotHt} - 0.0008 \times \texttt{CanHt} - 0.0391 \times \texttt{Dens} + 1.513$$

(Recall: sample mean $\overline{\texttt{LogLeafWt}} = 2.59$ is best constant predictor)

Example #2: Learned affine function

Solution to normal equations: $(0.1627, 0.1552, 0.3414, -0.0008, -0.0391, 1.513)$

Learned affine function:

$$f_{\text{ols}}(\texttt{Diam1},\texttt{Diam2},\texttt{TotHt},\texttt{CanHt},\texttt{Dens}) =$$
$$0.1627 \times \texttt{Diam1} + 0.1552 \times \texttt{Diam2} + 0.3414 \times \texttt{TotHt} - 0.0008 \times \texttt{CanHt} - 0.0391 \times \texttt{Dens} + 1.513$$

(Recall: sample mean $\overline{\texttt{LogLeafWt}} = 2.59$ is best constant predictor)

Training RMSE of f_{ols}:	0.168
Training RMSE of $\overline{\texttt{LogLeafWt}}$:	0.44

Example #2: Learned affine function

Solution to normal equations: $(0.1627, 0.1552, 0.3414, -0.0008, -0.0391, 1.513)$

Learned affine function:

$$f_{\text{ols}}(\texttt{Diam1},\texttt{Diam2},\texttt{TotHt},\texttt{CanHt},\texttt{Dens}) =$$
$$0.1627 \times \texttt{Diam1} + 0.1552 \times \texttt{Diam2} + 0.3414 \times \texttt{TotHt} - 0.0008 \times \texttt{CanHt} - 0.0391 \times \texttt{Dens} + 1.513$$

(Recall: sample mean $\overline{\texttt{LogLeafWt}} = 2.59$ is best constant predictor)

Training RMSE of f_{ols}:	0.168
Training RMSE of $\overline{\texttt{LogLeafWt}}$:	0.44

We have 20 additional test examples ...

Test RMSE of f_{ols}:	0.203
Test RMSE of $\overline{\texttt{LogLeafWt}}$:	0.32

Recap and discussion

- Restricting attention to linear functions allows us to very precisely characterize the minimizer of SSE training objective via normal equations
- Problem of minimizing SSE objective reduces to problem of solving linear equations

Features for Linear Regression

Feature engineering

Feature engineering: Deciding what to include in feature vector $\vec{x}$ (as numerical features)

Example: College GPA prediction

- Include `HS GPA`?
- Include `SAT Score`?
- Include **dummy variables** (e.g., $\mathbb{1}\{\texttt{Major} = \texttt{Physics}\}$)?
- ...

Feature engineering

Feature engineering: Deciding what to include in feature vector $\vec{x}$ (as numerical features)

Example: College GPA prediction

- Include `HS GPA`?
- Include `SAT Score`?
- Include **dummy variables** (e.g., $\mathbb{1}\{\texttt{Major} = \texttt{Physics}\}$)?
- ...

This requires some thought!

Feature engineering

Feature engineering: Deciding what to include in feature vector $\vec{x}$ (as numerical features)

Example: College GPA prediction

- Include `HS GPA`?
- Include `SAT Score`?
- Include **dummy variables** (e.g., $\mathbb{1}\{\texttt{Major} = \texttt{Physics}\}$)?
- ...

This requires some thought!

Some recommendations (Gelman and Hill, 2007):

1. "Include all input variables that, for substantive reasons, might be expected to be important in predicting the outcome"
2. "It is not always necessary to include these inputs as separate [features]—for example, sometimes several inputs can be averaged or summed to create a 'total score' that can be used as a single [feature] in the model"

Example #2: Logarithmic transformations

Task: Predict `LogLeafWt` from `Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`.

Example #2: Logarithmic transformations

Task: Predict `LogLeafWt` from `Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`.

- $10^{\texttt{LogLeafWt}}$ is a "weight", which may be proportional to some "volume"

Example #2: Logarithmic transformations

Task: Predict `LogLeafWt` from `Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`.

- $10^{\texttt{LogLeafWt}}$ is a "weight", which may be proportional to some "volume"
- Product `Diam1` $\times$ `Diam2` $\times$ `TotHt` can be regarded as a "volume"

Example #2: Logarithmic transformations

Task: Predict `LogLeafWt` from `Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`.

- $10^{\texttt{LogLeafWt}}$ is a "weight", which may be proportional to some "volume"
- Product $\texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt}$ can be regarded as a "volume"
- Perhaps

$$10^{\texttt{LogLeafWt}} \approx C \times \texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt} \ ?$$

  After taking $\log_{10}$ of both sides, this reads as

$$\texttt{LogLeafWt} \approx \log_{10}(C) + \log_{10}(\texttt{Diam1}) + \log_{10}(\texttt{Diam2}) + \log_{10}(\texttt{TotHt}) \ ?$$

Example #2: Logarithmic transformations

Task: Predict `LogLeafWt` from `Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`.

- $10^{\texttt{LogLeafWt}}$ is a "weight", which may be proportional to some "volume"
- Product $\texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt}$ can be regarded as a "volume"
- Perhaps

$$10^{\texttt{LogLeafWt}} \approx C \times \texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt} \ ?$$

  After taking $\log_{10}$ of both sides, this reads as

$$\texttt{LogLeafWt} \approx \log_{10}(C) + \log_{10}(\texttt{Diam1}) + \log_{10}(\texttt{Diam2}) + \log_{10}(\texttt{TotHt}) \ ?$$

Many different ways to use this!

Example #2: Logarithmic transformations

Task: Predict `LogLeafWt` from `Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`.

- $10^{\texttt{LogLeafWt}}$ is a "weight", which may be proportional to some "volume"
- Product $\texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt}$ can be regarded as a "volume"
- Perhaps

$$10^{\texttt{LogLeafWt}} \approx C \times \texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt} \ ?$$

  After taking $\log_{10}$ of both sides, this reads as

$$\texttt{LogLeafWt} \approx \log_{10}(C) + \log_{10}(\texttt{Diam1}) + \log_{10}(\texttt{Diam2}) + \log_{10}(\texttt{TotHt}) \ ?$$

Many different ways to use this!

- **Option 1:** Learn affine function of $\log_{10}(\texttt{Diam1})$, $\log_{10}(\texttt{Diam2})$, $\log_{10}(\texttt{TotHt})$

Example #2: Logarithmic transformations

Task: Predict `LogLeafWt` from `Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`.

- $10^{\texttt{LogLeafWt}}$ is a "weight", which may be proportional to some "volume"
- Product $\texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt}$ can be regarded as a "volume"
- Perhaps

$$10^{\texttt{LogLeafWt}} \approx C \times \texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt} \ ?$$

  After taking $\log_{10}$ of both sides, this reads as

$$\texttt{LogLeafWt} \approx \log_{10}(C) + \log_{10}(\texttt{Diam1}) + \log_{10}(\texttt{Diam2}) + \log_{10}(\texttt{TotHt}) \ ?$$

Many different ways to use this!

- **Option 1:** Learn affine function of $\log_{10}(\texttt{Diam1})$, $\log_{10}(\texttt{Diam2})$, $\log_{10}(\texttt{TotHt})$
- **Option 2:** Learn affine function of $\log_{10}(\texttt{Diam1}) + \log_{10}(\texttt{Diam2}) + \log_{10}(\texttt{TotHt})$
- ...

Example #2: Logarithmic transformations

Task: Predict `LogLeafWt` from `Diam1`, `Diam2`, `TotHt`, `CanHt`, `Dens`.

- $10^{\texttt{LogLeafWt}}$ is a "weight", which may be proportional to some "volume"
- Product $\texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt}$ can be regarded as a "volume"
- Perhaps

$$10^{\texttt{LogLeafWt}} \approx C \times \texttt{Diam1} \times \texttt{Diam2} \times \texttt{TotHt} \ ?$$

  After taking $\log_{10}$ of both sides, this reads as

$$\texttt{LogLeafWt} \approx \log_{10}(C) + \log_{10}(\texttt{Diam1}) + \log_{10}(\texttt{Diam2}) + \log_{10}(\texttt{TotHt}) \ ?$$

Many different ways to use this!

- **Option 1:** Learn affine function of $\log_{10}(\texttt{Diam1})$, $\log_{10}(\texttt{Diam2})$, $\log_{10}(\texttt{TotHt})$
- **Option 2:** Learn affine function of $\log_{10}(\texttt{Diam1}) + \log_{10}(\texttt{Diam2}) + \log_{10}(\texttt{TotHt})$
- ...

Which one will have better training RMSE? Which one will have better test RMSE?

Example #2: Learned affine functions of log-transformed features

Option 1:

$$0.6026 \times \log_{10}(\texttt{Diam1}) + 0.7519 \times \log_{10}(\texttt{Diam2}) + 0.9155 \times \log_{10}(\texttt{TotHt}) + 2.1482$$

Example #2: Learned affine functions of log-transformed features

Option 1:

$$0.6026 \times \log_{10}(\texttt{Diam1}) + 0.7519 \times \log_{10}(\texttt{Diam2}) + 0.9155 \times \log_{10}(\texttt{TotHt}) + 2.1482$$

Option 2:

$$0.7375 \times \big(\log_{10}(\texttt{Diam1}) + \log_{10}(\texttt{Diam2}) + \log_{10}(\texttt{TotHt})\big) + 2.1386$$

Example #2: Learned affine functions of log-transformed features

Option 1:

$$0.6026 \times \log_{10}(\texttt{Diam1}) + 0.7519 \times \log_{10}(\texttt{Diam2}) + 0.9155 \times \log_{10}(\texttt{TotHt}) + 2.1482$$

Option 2:

$$0.7375 \times \big(\log_{10}(\texttt{Diam1}) + \log_{10}(\texttt{Diam2}) + \log_{10}(\texttt{TotHt})\big) + 2.1386$$

	Original features	Option 1	Option 2
Training RMSE:	0.168	0.157	0.158
Test RMSE:	0.203	0.177	0.172

Upgrading linear regression

Feature expansion: Creating new feature vector $\vec{\varphi}(\vec{x})$ from existing features $\vec{x}$

Upgrading linear regression

Feature expansion: Creating new feature vector $\vec{\varphi}(\vec{x})$ from existing features $\vec{x}$

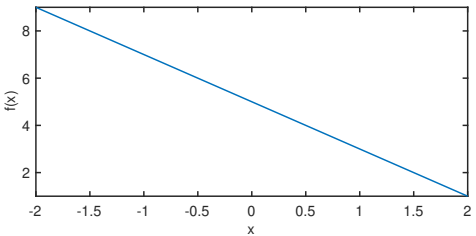
Examples: (Starting from single feature $x \in \mathbb{R}^1$)

1. (Affine expansion.) For $x \in \mathbb{R}$, define

$$\vec{\varphi}(x) := (x, 1) \in \mathbb{R}^2$$

Every affine function $f(x)$ can be written as $\vec{\varphi}(x) \cdot \vec{w}$ for some $\vec{w} \in \mathbb{R}^2$

$$\text{E.g.,} \quad f(x) \;=\; -2x + 5 \;=\; (x, 1) \cdot (-2, 5)$$



Upgrading linear regression

Feature expansion: Creating new feature vector $\vec{\varphi}(\vec{x})$ from existing features $\vec{x}$

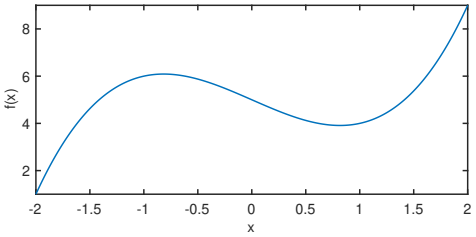
Examples: (Starting from single feature $x \in \mathbb{R}^1$)

2. (Polynomial expansion.) For $x \in \mathbb{R}$, define

$$\vec{\varphi}(x) := (1, x, x^2, \ldots, x^k) \in \mathbb{R}^{k+1}$$

Every degree-k univariate polynomial $f(x)$ can be written as $\vec{\varphi}(x) \cdot \vec{w}$ for some $\vec{w} \in \mathbb{R}^{k+1}$

$$\text{E.g.,} \quad f(x) \;=\; x^3 - 2x + 5 \;=\; (1, x, x^2, x^3) \cdot (5, -2, 0, 1)$$



Upgrading linear regression

Feature expansion: Creating new feature vector $\vec{\varphi}(\vec{x})$ from existing features $\vec{x}$

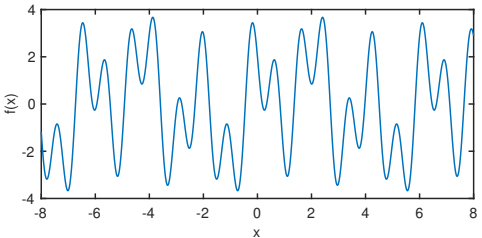
Examples: (Starting from single feature $x \in \mathbb{R}^1$)

3. (Trigonometric expansion.) For $x \in \mathbb{R}$, define

$$\vec{\varphi}(x) := (1, \sin(x), \cos(x), \sin(2x), \cos(2x), \ldots, \sin(kx), \cos(kx)) \in \mathbb{R}^{2k+1}$$

Every periodic function $f : \mathbb{R} \to \mathbb{R}$ (with period 2π) with harmonics up to k can be written as $f(x) = \vec{\varphi}(x) \cdot \vec{w}$ for some $\vec{w} \in \mathbb{R}^{2k+1}$

E.g., $\quad f(x) = \sin(x) + 2\cos(3x) - 2\sin(7x)$



Upgrading linear regression

Feature expansion: Creating new feature vector $\vec{\varphi}(\vec{x})$ from existing features $\vec{x}$
Examples: (Starting from d features $\vec{x} \in \mathbb{R}^d$)

4. (Interaction features.) For $\vec{x} \in \mathbb{R}^d$, define

$$\vec{\varphi}(x) := (x_i x_j : 1 \leq i < j \leq d) \in \mathbb{R}^{\binom{d}{2}}$$

Upgrading linear regression

Feature expansion: Creating new feature vector $\vec{\varphi}(\vec{x})$ from existing features $\vec{x}$
Examples: (Starting from d features $\vec{x} \in \mathbb{R}^d$)

4. (Interaction features.) For $\vec{x} \in \mathbb{R}^d$, define

$$\vec{\varphi}(x) := (x_i x_j : 1 \leq i < j \leq d) \in \mathbb{R}^{\binom{d}{2}}$$

- Terms $x_i x_j$ are called **(pairwise) interaction features**
  If each feature takes values only in $\{0, 1\}$, then

$$x_i x_j = \begin{cases} 1 & \text{if } x_i = 1 \text{ AND } x_j = 1 \\ 0 & \text{otherwise} \end{cases}$$

Upgrading linear regression

Feature expansion: Creating new feature vector $\vec{\varphi}(\vec{x})$ from existing features $\vec{x}$
Examples: (Starting from d features $\vec{x} \in \mathbb{R}^d$)

4. (Interaction features.) For $\vec{x} \in \mathbb{R}^d$, define

$$\vec{\varphi}(x) := (x_i x_j : 1 \leq i < j \leq d) \in \mathbb{R}^{\binom{d}{2}}$$

- Terms $x_i x_j$ are called **(pairwise) interaction features**
  If each feature takes values only in $\{0, 1\}$, then

$$x_i x_j = \begin{cases} 1 & \text{if } x_i = 1 \text{ AND } x_j = 1 \\ 0 & \text{otherwise} \end{cases}$$

- Can generalize to k-way interactions, increasing total number of features to $\binom{d}{k}$
  (**Caution:** $\binom{d}{k}$ grows exponentially with k)

Upgrading linear regression

Feature expansion: Creating new feature vector $\vec{\varphi}(\vec{x})$ from existing features $\vec{x}$

Examples: (Starting from d features $\vec{x} \in \mathbb{R}^d$)

5. (Model averaging.) Suppose you have M predictors $f_1, f_2, \ldots, f_M$ (obtained earlier ...)
   For $\vec{x} \in \mathbb{R}^d$, define
   $$\vec{\varphi}(\vec{x}) := \big(f_1(\vec{x}), f_2(\vec{x}), \ldots, f_M(\vec{x})\big) \in \mathbb{R}^M$$
   Uniform model averaging is $f_{\text{avg}}(\vec{x}) = \vec{\varphi}(\vec{x}) \cdot \vec{w}$ where
   $$w_i = \frac{1}{M} \quad \text{for all } i = 1, \ldots, M$$
   To have "averaging" interpretation in general, constrain $\vec{w}$ s.t. all $w_i \geq 0$ and $\sum_{i=1}^{M} w_i = 1$

Scaling features

OLS is **scale invariant:**

- Suppose weight vector $\vec{w}$ satisfies normal equations
- If we replace j-th feature x_j with $x_j/2$ (in all training examples), then same weight vector as before except with w_j replaced by $2w_j$ satisfies the new normal equations

(In fact, invariant to arbitrary "change-of-basis")

Scaling features

OLS is **scale invariant:**

- Suppose weight vector $\vec{w}$ satisfies normal equations
- If we replace j-th feature x_j with $x_j/2$ (in all training examples), then same weight vector as before except with w_j replaced by $2w_j$ satisfies the new normal equations

(In fact, invariant to arbitrary "change-of-basis")

Caution:

- Algorithms for solving linear equations use floating point numbers with limited numerical precision

Scaling features

OLS is **scale invariant:**

- Suppose weight vector $\vec{w}$ satisfies normal equations
- If we replace j-th feature x_j with $x_j/2$ (in all training examples), then same weight vector as before except with w_j replaced by $2w_j$ satisfies the new normal equations

(In fact, invariant to arbitrary "change-of-basis")

Caution:

- Algorithms for solving linear equations use floating point numbers with limited numerical precision
  - Common strategy: scale features so that every feature has same (empirical) stddev
  - If also using "intercept" coefficient, then also common to **standardize** the features:
    I.e., perform shifting and scaling so every feature has (empirical) mean zero and unit stddev

Scaling features

OLS is **scale invariant:**

- Suppose weight vector $\vec{w}$ satisfies normal equations
- If we replace j-th feature x_j with $x_j/2$ (in all training examples), then same weight vector as before except with w_j replaced by $2w_j$ satisfies the new normal equations

(In fact, invariant to arbitrary "change-of-basis")

Caution:

- Algorithms for solving linear equations use floating point numbers with limited numerical precision
  - Common strategy: scale features so that every feature has same (empirical) stddev
  - If also using "intercept" coefficient, then also common to **standardize** the features: I.e., perform shifting and scaling so every feature has (empirical) mean zero and unit stddev
- Many other algorithms for linear regression are not scale invariant!

Too many features?

Suppose degree-k polynomial expansion of a scalar feature is used:

$$\vec{\varphi}(x) = (1, x, x^2, \ldots, x^k) \in \mathbb{R}^{k+1}$$

- Number of features is $d = k + 1$

Too many features?

Suppose degree-k polynomial expansion of a scalar feature is used:

$$\vec{\varphi}(x) = (1, x, x^2, \ldots, x^k) \in \mathbb{R}^{k+1}$$

- Number of features is $d = k + 1$
- **Fact:** Any function of $\leq k + 1$ points can be interpolated by a polynomial of degree $\leq k$

Too many features?

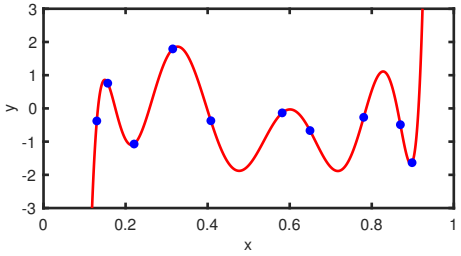
Suppose degree-k polynomial expansion of a scalar feature is used:

$$\vec{\varphi}(x) = (1, x, x^2, \ldots, x^k) \in \mathbb{R}^{k+1}$$

- Number of features is $d = k + 1$
- **Fact:** Any function of $\leq k + 1$ points can be interpolated by a polynomial of degree $\leq k$
- So, if $n \leq k + 1 = d$, then OLS on $(\vec{\varphi}(x_1), y_1), \ldots, (\vec{\varphi}(x_n), y_n)$ returns $\vec{w}_{\text{ols}} \in \mathbb{R}^d$ such that

$$\vec{\varphi}(x_i) \cdot \vec{w}_{\text{ols}} = y_i \quad \text{for all } i = 1, \ldots, n$$

(assuming no two training examples have same feature vector but different labels)



Too many features?

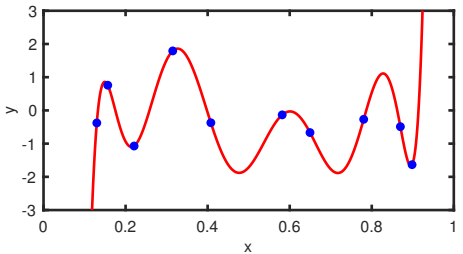
Suppose degree-k polynomial expansion of a scalar feature is used:

$$\vec{\varphi}(x) = (1, x, x^2, \ldots, x^k) \in \mathbb{R}^{k+1}$$

- Number of features is $d = k + 1$
- **Fact:** Any function of $\leq k + 1$ points can be interpolated by a polynomial of degree $\leq k$
- So, if $n \leq k + 1 = d$, then OLS on $(\vec{\varphi}(x_1), y_1), \ldots, (\vec{\varphi}(x_n), y_n)$ returns $\vec{w}_{\text{ols}} \in \mathbb{R}^d$ such that

$$\vec{\varphi}(x_i) \cdot \vec{w}_{\text{ols}} = y_i \quad \text{for all } i = 1, \ldots, n$$

(assuming no two training examples have same feature vector but different labels)



SSE on training data is 0, regardless of actual quality of $f(x) = \vec{\varphi}(x) \cdot \vec{w}_{\text{ols}}$

Too many features?

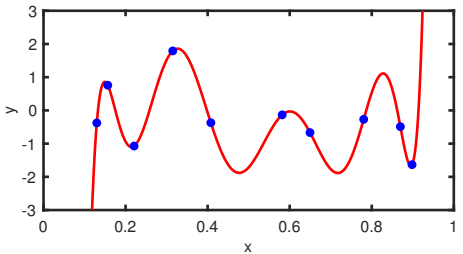
Suppose degree-k polynomial expansion of a scalar feature is used:

$$\vec{\varphi}(x) = (1, x, x^2, \ldots, x^k) \in \mathbb{R}^{k+1}$$

- Number of features is $d = k + 1$
- **Fact:** Any function of $\leq k + 1$ points can be interpolated by a polynomial of degree $\leq k$
- So, if $n \leq k + 1 = d$, then OLS on $(\vec{\varphi}(x_1), y_1), \ldots, (\vec{\varphi}(x_n), y_n)$ returns $\vec{w}_{\text{ols}} \in \mathbb{R}^d$ such that

$$\vec{\varphi}(x_i) \cdot \vec{w}_{\text{ols}} = y_i \quad \text{for all } i = 1, \ldots, n$$

(assuming no two training examples have same feature vector but different labels)



SSE on training data is 0, regardless of actual quality of $f(x) = \vec{\varphi}(x) \cdot \vec{w}_{\text{ols}}$

Potential for "over-fitting" with OLS

Linear Algebraic View of OLS

Motivation

Questions:

- Can there be multiple solutions to normal equations?
  (If so, OLS depends on method of solving equations)
- Can there be no solutions?
  (If so, algorithm for OLS might fail!)

Matrix notation for OLS

Suppose the training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$

Matrix notation for OLS

Suppose the training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$
Arrange feature vectors and labels in $n \times d$ matrix A and $n \times 1$ (column) vector $\vec{b}$:

$$A := \begin{bmatrix} \longleftarrow & \vec{x}_1^{\mathsf{T}} & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{x}_n^{\mathsf{T}} & \longrightarrow \end{bmatrix} \in \mathbb{R}^{n \times d}, \qquad \vec{b} := \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

(By default, vectors $\vec{x} \in \mathbb{R}^d$ are regarded as column vectors; so write $\vec{x}^{\mathsf{T}}$ to denote a row vector)

Matrix notation for OLS

Suppose the training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$
Arrange feature vectors and labels in $n \times d$ matrix A and $n \times 1$ (column) vector $\vec{b}$:

$$A := \begin{bmatrix} \longleftarrow & \vec{x}_1^{\mathsf{T}} & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{x}_n^{\mathsf{T}} & \longrightarrow \end{bmatrix} \in \mathbb{R}^{n \times d}, \qquad \vec{b} := \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

(By default, vectors $\vec{x} \in \mathbb{R}^d$ are regarded as column vectors; so write $\vec{x}^{\mathsf{T}}$ to denote a row vector)

- For any $\vec{w} \in \mathbb{R}^d$, matrix-vector product $A\vec{w}$ is a (column) vector in $\mathbb{R}^n$, and so is $A\vec{w} - \vec{b}$

Matrix notation for OLS

Suppose the training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$
Arrange feature vectors and labels in $n \times d$ matrix A and $n \times 1$ (column) vector $\vec{b}$:

$$A := \begin{bmatrix} \longleftarrow & \vec{x}_1^{\mathsf{T}} & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{x}_n^{\mathsf{T}} & \longrightarrow \end{bmatrix} \in \mathbb{R}^{n \times d}, \qquad \vec{b} := \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

(By default, vectors $\vec{x} \in \mathbb{R}^d$ are regarded as column vectors; so write $\vec{x}^{\mathsf{T}}$ to denote a row vector)

- For any $\vec{w} \in \mathbb{R}^d$, matrix-vector product $A\vec{w}$ is a (column) vector in $\mathbb{R}^n$, and so is $A\vec{w} - \vec{b}$
- Squared Euclidean length of $A\vec{w} - \vec{b}$ is

$$\|A\vec{w} - \vec{b}\|_2^2 = \sum_{i=1}^{n} (\vec{x}_i \cdot \vec{w} - y_i)^2 = \mathrm{sse}(\vec{w}; \mathcal{S})$$

Matrix notation for OLS

Suppose the training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$
Arrange feature vectors and labels in $n \times d$ matrix A and $n \times 1$ (column) vector $\vec{b}$:

$$A := \begin{bmatrix} \longleftarrow & \vec{x}_1^{\mathsf{T}} & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{x}_n^{\mathsf{T}} & \longrightarrow \end{bmatrix} \in \mathbb{R}^{n \times d}, \qquad \vec{b} := \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

(By default, vectors $\vec{x} \in \mathbb{R}^d$ are regarded as column vectors; so write $\vec{x}^{\mathsf{T}}$ to denote a row vector)

- For any $\vec{w} \in \mathbb{R}^d$, matrix-vector product $A\vec{w}$ is a (column) vector in $\mathbb{R}^n$, and so is $A\vec{w} - \vec{b}$
- Squared Euclidean length of $A\vec{w} - \vec{b}$ is

$$\|A\vec{w} - \vec{b}\|_2^2 = \sum_{i=1}^{n} (\vec{x}_i \cdot \vec{w} - y_i)^2 = \mathrm{sse}(\vec{w}; \mathcal{S})$$

- Normal equations in matrix form:

$$(A^{\mathsf{T}} A)\vec{w} = A^{\mathsf{T}} \vec{b},$$

with $A^{\mathsf{T}} A \in \mathbb{R}^{d \times d}$, $A^{\mathsf{T}} \vec{b} \in \mathbb{R}^d$, and vector of d unknown variables $\vec{w}$

Linear algebra of OLS: Invertible case

If $A^\mathsf{T}A$ is invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have a unique solution, given by

$$\vec{w} = (A^\mathsf{T}A)^{-1}A^\mathsf{T}\vec{b}$$

Linear algebra of OLS: Invertible case

If $A^\mathsf{T}A$ is invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have a unique solution, given by

$$\vec{w} = (A^\mathsf{T}A)^{-1}A^\mathsf{T}\vec{b}$$

Linear algebra of OLS: Non-invertible case

If $A^\intercal A$ is not invertible, then the normal equations

$$(A^\intercal A)\vec{w} = A^\intercal \vec{b}$$

have infinitely-many solutions or no solutions

Linear algebra of OLS: Non-invertible case

If $A^\mathsf{T}A$ is not invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have infinitely-many solutions ~~or no solutions~~ (. . . there is always a solution!)

Fact. For any $A \in \mathbb{R}^{n \times d}$ and $\vec{b} \in \mathbb{R}^n$, there is a unique way to write $\vec{b} = \vec{p} + \vec{r}$, where

- $\vec{p}$ is in the column space of A
- $\vec{r}$ is in the null space of A^T

(In particular, $\vec{p}$ and $\vec{r}$ are orthogonal)

Linear algebra of OLS: Non-invertible case

If $A^\mathsf{T}A$ is not invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have infinitely-many solutions ~~or no solutions~~ (. . . there is always a solution!)

Fact. For any $A \in \mathbb{R}^{n \times d}$ and $\vec{b} \in \mathbb{R}^n$, there is a unique way to write $\vec{b} = \vec{p} + \vec{r}$, where

- $\vec{p}$ is in the column space of A
- $\vec{r}$ is in the null space of A^T

(In particular, $\vec{p}$ and $\vec{r}$ are orthogonal)

- Write $\vec{b} = \vec{p} + \vec{r}$ as above

Linear algebra of OLS: Non-invertible case

If $A^\mathsf{T}A$ is not invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have infinitely-many solutions ~~or no solutions~~ (. . . there is always a solution!)

Fact. For any $A \in \mathbb{R}^{n\times d}$ and $\vec{b} \in \mathbb{R}^n$, there is a unique way to write $\vec{b} = \vec{p} + \vec{r}$, where

- $\vec{p}$ is in the column space of A
- $\vec{r}$ is in the null space of A^T

(In particular, $\vec{p}$ and $\vec{r}$ are orthogonal)

- Write $\vec{b} = \vec{p} + \vec{r}$ as above
- Since $\vec{p}$ is in column space of A, there is some $\vec{x} \in \mathbb{R}^d$ s.t. $\vec{p} = A\vec{x}$

Linear algebra of OLS: Non-invertible case

If $A^\mathsf{T}A$ is not invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have infinitely-many solutions ~~or no solutions~~ (. . . there is always a solution!)

Fact. For any $A \in \mathbb{R}^{n\times d}$ and $\vec{b} \in \mathbb{R}^n$, there is a unique way to write $\vec{b} = \vec{p} + \vec{r}$, where

- $\vec{p}$ is in the column space of A
- $\vec{r}$ is in the null space of A^T

(In particular, $\vec{p}$ and $\vec{r}$ are orthogonal)

- Write $\vec{b} = \vec{p} + \vec{r}$ as above
- Since $\vec{p}$ is in column space of A, there is some $\vec{x} \in \mathbb{R}^d$ s.t. $\vec{p} = A\vec{x}$

$$A^\mathsf{T}\vec{b}$$

Linear algebra of OLS: Non-invertible case

If $A^\mathsf{T}A$ is not invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have infinitely-many solutions ~~or no solutions~~ (. . . there is always a solution!)

Fact. For any $A \in \mathbb{R}^{n\times d}$ and $\vec{b} \in \mathbb{R}^n$, there is a unique way to write $\vec{b} = \vec{p} + \vec{r}$, where

- $\vec{p}$ is in the column space of A
- $\vec{r}$ is in the null space of A^T

(In particular, $\vec{p}$ and $\vec{r}$ are orthogonal)

- Write $\vec{b} = \vec{p} + \vec{r}$ as above
- Since $\vec{p}$ is in column space of A, there is some $\vec{x} \in \mathbb{R}^d$ s.t. $\vec{p} = A\vec{x}$

$$A^\mathsf{T}\vec{b} = A^\mathsf{T}(\vec{p} + \vec{r})$$

Linear algebra of OLS: Non-invertible case

If $A^\mathsf{T}A$ is not invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have infinitely-many solutions ~~or no solutions~~ (. . . there is always a solution!)

Fact. For any $A \in \mathbb{R}^{n\times d}$ and $\vec{b} \in \mathbb{R}^n$, there is a unique way to write $\vec{b} = \vec{p} + \vec{r}$, where

- $\vec{p}$ is in the column space of A
- $\vec{r}$ is in the null space of A^T

(In particular, $\vec{p}$ and $\vec{r}$ are orthogonal)

- Write $\vec{b} = \vec{p} + \vec{r}$ as above
- Since $\vec{p}$ is in column space of A, there is some $\vec{x} \in \mathbb{R}^d$ s.t. $\vec{p} = A\vec{x}$

$$A^\mathsf{T}\vec{b} = A^\mathsf{T}(\vec{p} + \vec{r}) = A^\mathsf{T}(A\vec{x} + \vec{r})$$

Linear algebra of OLS: Non-invertible case

If $A^\mathsf{T}A$ is not invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have infinitely-many solutions ~~or no solutions~~ (. . . there is always a solution!)

Fact. For any $A \in \mathbb{R}^{n\times d}$ and $\vec{b} \in \mathbb{R}^n$, there is a unique way to write $\vec{b} = \vec{p} + \vec{r}$, where

- $\vec{p}$ is in the column space of A
- $\vec{r}$ is in the null space of A^T

(In particular, $\vec{p}$ and $\vec{r}$ are orthogonal)

- Write $\vec{b} = \vec{p} + \vec{r}$ as above
- Since $\vec{p}$ is in column space of A, there is some $\vec{x} \in \mathbb{R}^d$ s.t. $\vec{p} = A\vec{x}$

$$A^\mathsf{T}\vec{b} = A^\mathsf{T}(\vec{p} + \vec{r}) = A^\mathsf{T}(A\vec{x} + \vec{r}) = A^\mathsf{T}(A\vec{x}) + A^\mathsf{T}\vec{r}$$

Linear algebra of OLS: Non-invertible case

If $A^\mathsf{T}A$ is not invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have infinitely-many solutions ~~or no solutions~~ (. . . there is always a solution!)

Fact. For any $A \in \mathbb{R}^{n\times d}$ and $\vec{b} \in \mathbb{R}^n$, there is a unique way to write $\vec{b} = \vec{p} + \vec{r}$, where

- $\vec{p}$ is in the column space of A
- $\vec{r}$ is in the null space of A^T

(In particular, $\vec{p}$ and $\vec{r}$ are orthogonal)

- Write $\vec{b} = \vec{p} + \vec{r}$ as above
- Since $\vec{p}$ is in column space of A, there is some $\vec{x} \in \mathbb{R}^d$ s.t. $\vec{p} = A\vec{x}$

$$A^\mathsf{T}\vec{b} = A^\mathsf{T}(\vec{p} + \vec{r}) = A^\mathsf{T}(A\vec{x} + \vec{r}) = A^\mathsf{T}(A\vec{x}) + A^\mathsf{T}\vec{r} = (A^\mathsf{T}A)\vec{x} + \vec{0}$$

Linear algebra of OLS: Non-invertible case

If $A^\mathsf{T}A$ is not invertible, then the normal equations

$$(A^\mathsf{T}A)\vec{w} = A^\mathsf{T}\vec{b}$$

have infinitely-many solutions ~~or no solutions~~ (. . . there is always a solution!)

Fact. For any $A \in \mathbb{R}^{n\times d}$ and $\vec{b} \in \mathbb{R}^n$, there is a unique way to write $\vec{b} = \vec{p} + \vec{r}$, where

- $\vec{p}$ is in the column space of A
- $\vec{r}$ is in the null space of A^T

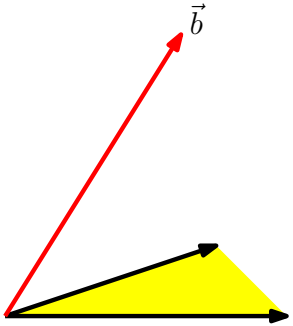
(In particular, $\vec{p}$ and $\vec{r}$ are orthogonal)

- Write $\vec{b} = \vec{p} + \vec{r}$ as above
- Since $\vec{p}$ is in column space of A, there is some $\vec{x} \in \mathbb{R}^d$ s.t. $\vec{p} = A\vec{x}$

$$A^\mathsf{T}\vec{b} = A^\mathsf{T}(\vec{p} + \vec{r}) = A^\mathsf{T}(A\vec{x} + \vec{r}) = A^\mathsf{T}(A\vec{x}) + A^\mathsf{T}\vec{r} = (A^\mathsf{T}A)\vec{x} + \vec{0}$$

so $\vec{w} := \vec{x}$ satisfies normal equations

Linear algebra of OLS: Orthogonal projection

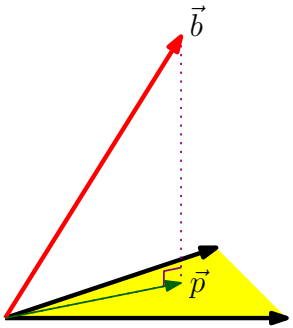


Column space of A is
$\{A\vec{w} : \vec{w} \in \mathbb{R}^d\} \subseteq \mathbb{R}^n$

Let's interpret $\min_{\vec{w} \in \mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$ as

"Find vector in column space of A that is closest to $\vec{b}$"

Linear algebra of OLS: Orthogonal projection



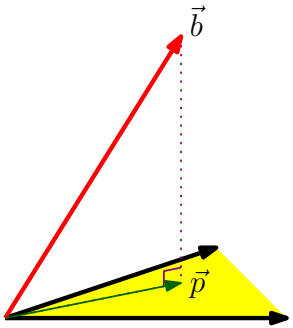
Column space of A is
$\{A\vec{w} : \vec{w} \in \mathbb{R}^d\} \subseteq \mathbb{R}^n$

Let's interpret $\min_{\vec{w} \in \mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$ as

"Find vector in column space of A that is closest to $\vec{b}$"

- Solution is vector $\vec{p}$ such that $\vec{r} := \vec{b} - \vec{p}$ is orthogonal to $\vec{p}$

Linear algebra of OLS: Orthogonal projection



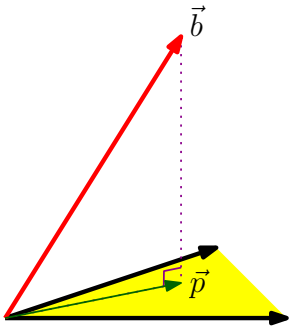
Column space of A is
$\{A\vec{w} : \vec{w} \in \mathbb{R}^d\} \subseteq \mathbb{R}^n$

Let's interpret $\min_{\vec{w} \in \mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$ as

"Find vector in column space of A that is closest to $\vec{b}$"

- Solution is vector $\vec{p}$ such that $\vec{r} := \vec{b} - \vec{p}$ is orthogonal to $\vec{p}$
- Vector $\vec{p}$ is called the **orthogonal projection** of $\vec{b}$ onto column space of A

Linear algebra of OLS: Orthogonal projection



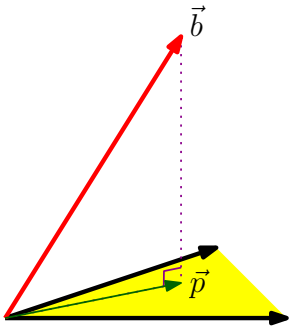
Column space of A is
$\{A\vec{w} : \vec{w} \in \mathbb{R}^d\} \subseteq \mathbb{R}^n$

Let's interpret $\min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$ as

"Find vector in column space of A that is closest to $\vec{b}$"

- Solution is vector $\vec{p}$ such that $\vec{r} := \vec{b} - \vec{p}$ is orthogonal to $\vec{p}$
- Vector $\vec{p}$ is called the **orthogonal projection** of $\vec{b}$ onto column space of A
- Vector $\vec{r}$ is called the **residual**, and $\|\vec{r}\|_2^2 = \min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$

Linear algebra of OLS: Orthogonal projection



Column space of A is $\{A\vec{w} : \vec{w} \in \mathbb{R}^d\} \subseteq \mathbb{R}^n$

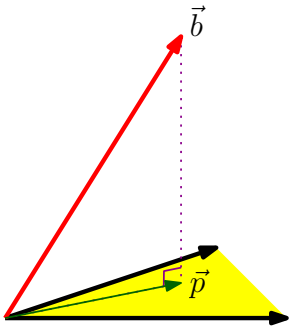
Let's interpret $\min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$ as

"Find vector in column space of A that is closest to $\vec{b}$"

- Solution is vector $\vec{p}$ such that $\vec{r} := \vec{b} - \vec{p}$ is orthogonal to $\vec{p}$
- Vector $\vec{p}$ is called the **orthogonal projection** of $\vec{b}$ onto column space of A
- Vector $\vec{r}$ is called the **residual**, and $\|\vec{r}\|_2^2 = \min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$

Question: How can there be multiple solutions $\vec{w}$?

Linear algebra of OLS: Orthogonal projection



Column space of A is $\{A\vec{w} : \vec{w} \in \mathbb{R}^d\} \subseteq \mathbb{R}^n$

Let's interpret $\min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$ as

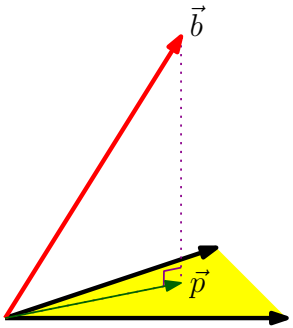
"Find vector in column space of A that is closest to $\vec{b}$"

- Solution is vector $\vec{p}$ such that $\vec{r} := \vec{b} - \vec{p}$ is orthogonal to $\vec{p}$
- Vector $\vec{p}$ is called the **orthogonal projection** of $\vec{b}$ onto column space of A
- Vector $\vec{r}$ is called the **residual**, and $\|\vec{r}\|_2^2 = \min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$

Question: How can there be multiple solutions $\vec{w}$?

- Orthogonal projection $\vec{p}$ of $\vec{b}$ is unique, but there could be multiple $\vec{w} \in \mathbb{R}^d$ such that $\vec{p} = A\vec{w}$

Linear algebra of OLS: Orthogonal projection



Column space of A is $\{A\vec{w} : \vec{w} \in \mathbb{R}^d\} \subseteq \mathbb{R}^n$

Let's interpret $\min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$ as

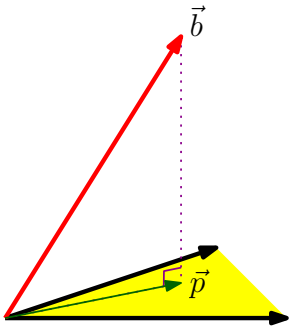
"Find vector in column space of A that is closest to $\vec{b}$"

- Solution is vector $\vec{p}$ such that $\vec{r} := \vec{b} - \vec{p}$ is orthogonal to $\vec{p}$
- Vector $\vec{p}$ is called the **orthogonal projection** of $\vec{b}$ onto column space of A
- Vector $\vec{r}$ is called the **residual**, and $\|\vec{r}\|_2^2 = \min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$

Question: How can there be multiple solutions $\vec{w}$?

- Orthogonal projection $\vec{p}$ of $\vec{b}$ is unique, but there could be multiple $\vec{w} \in \mathbb{R}^d$ such that $\vec{p} = A\vec{w}$
- [Only one such $\vec{w}$] $\Leftrightarrow$ [A has linearly independent columns] $\Leftrightarrow$ [A has rank d]

Linear algebra of OLS: Orthogonal projection



Column space of A is $\{A\vec{w} : \vec{w} \in \mathbb{R}^d\} \subseteq \mathbb{R}^n$

Let's interpret $\min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$ as

"Find vector in column space of A that is closest to $\vec{b}$"

- Solution is vector $\vec{p}$ such that $\vec{r} := \vec{b} - \vec{p}$ is orthogonal to $\vec{p}$
- Vector $\vec{p}$ is called the **orthogonal projection** of $\vec{b}$ onto column space of A
- Vector $\vec{r}$ is called the **residual**, and $\|\vec{r}\|_2^2 = \min_{\vec{w}\in\mathbb{R}^d} \|A\vec{w} - \vec{b}\|_2^2$

Question: How can there be multiple solutions $\vec{w}$?

- Orthogonal projection $\vec{p}$ of $\vec{b}$ is unique, but there could be multiple $\vec{w} \in \mathbb{R}^d$ such that $\vec{p} = A\vec{w}$
- [Only one such $\vec{w}$] $\Leftrightarrow$ [A has linearly independent columns] $\Leftrightarrow$ [A has rank d]
- If $n < d$, then A has rank $< d$, and hence infinitely many solutions to normal equations!

Recap and discussion

Linear algebra reveals source of potential "instability" in OLS

- Minimizer is not necessarily unique (and definitely not unique if $n < d$)
- May need other approaches to linear regression beyond OLS

Classical Statistics View of OLS

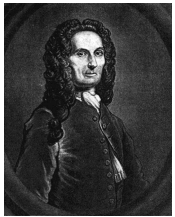
Motivation

Classical statistics view of OLS:

- Under what assumptions is the linear function that minimizes the SSE on training data any good?
- What are further implications of these assumptions?

Aside: Normal distribution

$\mathrm{N}(\mu, \sigma^2)$ denotes the **normal** (a.k.a. **Gaussian**) distribution with mean parameter $\mu \in \mathbb{R}$ and variance parameter $\sigma^2 > 0$



Abraham de Moivre, 1738

Carl Friedrich Gauss, 1823

Aside: Normal distribution

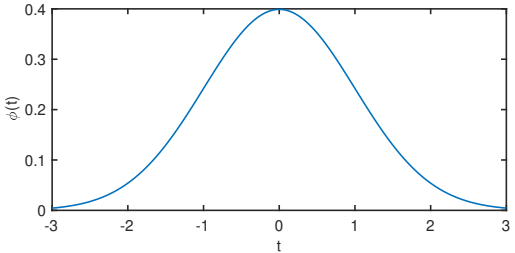
$\mathrm{N}(\mu, \sigma^2)$ denotes the **normal** (a.k.a. **Gaussian**) distribution with mean parameter $\mu \in \mathbb{R}$ and variance parameter $\sigma^2 > 0$

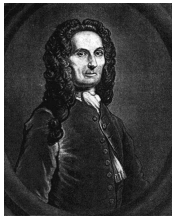
If $Y \sim \mathrm{N}(\mu, \sigma^2)$, then

$$\mathbb{E}(Y) = \mu, \quad \mathrm{var}(Y) = \sigma^2$$

and $(Y - \mu)/\sigma$ has probability density function given by

$$\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$$





Abraham de Moivre, 1738

Carl Friedrich Gauss, 1823

Statistical model for linear regression

Statistical model used to motivate OLS:

Normal linear regression model:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID, and for any $\vec{x} \in \mathbb{R}^d$,

$$(Y \mid \vec{X} = \vec{x}) \sim \mathrm{N}(\vec{x} \cdot \vec{w}, \sigma^2)$$

Last part says: "Conditional distribution of Y given $\vec{X} = \vec{x}$ is normal with mean $\vec{x} \cdot \vec{w}$ and variance σ^2"

Statistical model for linear regression

Statistical model used to motivate OLS:

Normal linear regression model:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID, and for any $\vec{x} \in \mathbb{R}^d$,

$$(Y \mid \vec{X} = \vec{x}) \sim \mathrm{N}(\vec{x} \cdot \vec{w}, \sigma^2)$$

Last part says: "Conditional distribution of Y given $\vec{X} = \vec{x}$ is normal with mean $\vec{x} \cdot \vec{w}$ and variance σ^2"

- $\vec{w} \in \mathbb{R}^d$ is a **parameter** of the statistical model
- $\sigma^2 > 0$ is also a parameter of the model
- Marginal distribution of $\vec{X}$ unspecified; doesn't involve $\vec{w}$ or σ^2

Statistical model for linear regression

Statistical model used to motivate OLS:

Normal linear regression model:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID, and for any $\vec{x} \in \mathbb{R}^d$,

$$(Y \mid \vec{X} = \vec{x}) \sim \mathrm{N}(\vec{x} \cdot \vec{w}, \sigma^2)$$

Last part says: "Conditional distribution of Y given $\vec{X} = \vec{x}$ is normal with mean $\vec{x} \cdot \vec{w}$ and variance σ^2"

- $\vec{w} \in \mathbb{R}^d$ is a **parameter** of the statistical model
- $\sigma^2 > 0$ is also a parameter of the model
- Marginal distribution of $\vec{X}$ unspecified; doesn't involve $\vec{w}$ or σ^2

Observation: In this model, $\mathbb{E}[Y \mid \vec{X} = \vec{x}] = \vec{x} \cdot \vec{w}$, a linear function!

Statistical model for linear regression

Statistical model used to motivate OLS:

Normal linear regression model:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID, and for any $\vec{x} \in \mathbb{R}^d$,

$$(Y \mid \vec{X} = \vec{x}) \sim \mathrm{N}(\vec{x} \cdot \vec{w}, \sigma^2)$$

Last part says: "Conditional distribution of Y given $\vec{X} = \vec{x}$ is normal with mean $\vec{x} \cdot \vec{w}$ and variance σ^2"

- $\vec{w} \in \mathbb{R}^d$ is a **parameter** of the statistical model
- $\sigma^2 > 0$ is also a parameter of the model
- Marginal distribution of $\vec{X}$ unspecified; doesn't involve $\vec{w}$ or σ^2

Observation: In this model, $\mathbb{E}[Y \mid \vec{X} = \vec{x}] = \vec{x} \cdot \vec{w}$, a linear function!

- In this model with specific parameter $\vec{w}$, predictor of smallest MSE is $f(\vec{x}) = \vec{x} \cdot \vec{w}$

Statistical model for linear regression

Statistical model used to motivate OLS:

Normal linear regression model:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID, and for any $\vec{x} \in \mathbb{R}^d$,

$$(Y \mid \vec{X} = \vec{x}) \sim \mathrm{N}(\vec{x} \cdot \vec{w}, \sigma^2)$$

Last part says: "Conditional distribution of Y given $\vec{X} = \vec{x}$ is normal with mean $\vec{x} \cdot \vec{w}$ and variance σ^2"

- $\vec{w} \in \mathbb{R}^d$ is a **parameter** of the statistical model
- $\sigma^2 > 0$ is also a parameter of the model
- Marginal distribution of $\vec{X}$ unspecified; doesn't involve $\vec{w}$ or σ^2

Observation: In this model, $\mathbb{E}[Y \mid \vec{X} = \vec{x}] = \vec{x} \cdot \vec{w}$, a linear function!

- In this model with specific parameter $\vec{w}$, predictor of smallest MSE is $f(\vec{x}) = \vec{x} \cdot \vec{w}$
- If you believe this model *for some (unknown) parameter* $\vec{w}$, estimate $\vec{w}$ using training data!

MLE for normal linear regression model

Suppose training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$

Recall: Under normal linear regression model with parameters $\vec{w}$ and σ^2,

$$(Y_i \mid \vec{X}_i = \vec{x}_i) \sim \mathrm{N}(\vec{x}_i \cdot \vec{w}, \sigma^2)$$

MLE for normal linear regression model

Suppose training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$

Recall: Under normal linear regression model with parameters $\vec{w}$ and σ^2,

$$(Y_i \mid \vec{X}_i = \vec{x}_i) \sim \mathrm{N}(\vec{x}_i \cdot \vec{w}, \sigma^2)$$

Likelihood of $(\vec{w}, \sigma^2)$: Using fact that training examples are treated as IID,

$$L(\vec{w}, \sigma^2) \;=\; p_{\vec{w},\sigma^2}((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)) \;\propto\; \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \vec{x}_i \cdot \vec{w})^2}{2\sigma^2}\right)$$

(constant of proportionality doesn't depend on $\vec{w}$ or σ^2)

MLE for normal linear regression model

Suppose training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$

Recall: Under normal linear regression model with parameters $\vec{w}$ and σ^2,

$$(Y_i \mid \vec{X}_i = \vec{x}_i) \sim \mathrm{N}(\vec{x}_i \cdot \vec{w}, \sigma^2)$$

Likelihood of $(\vec{w}, \sigma^2)$: Using fact that training examples are treated as IID,

$$L(\vec{w}, \sigma^2) \;=\; p_{\vec{w},\sigma^2}((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)) \;\propto\; \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \vec{x}_i \cdot \vec{w})^2}{2\sigma^2}\right)$$

(constant of proportionality doesn't depend on $\vec{w}$ or σ^2)

Log-likelihood of $(\vec{w}, \sigma^2)$:

$$\ln L(\vec{w}, \sigma^2) \;=\; -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \vec{x}_i \cdot \vec{w})^2 \;+\; \frac{n}{2} \ln \frac{1}{\sigma^2} \;+\; (\text{terms that don't depend on } \vec{w} \text{ or } \sigma^2)$$

MLE for normal linear regression model

Suppose training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$

Recall: Under normal linear regression model with parameters $\vec{w}$ and σ^2,

$$(Y_i \mid \vec{X}_i = \vec{x}_i) \sim \mathrm{N}(\vec{x}_i \cdot \vec{w}, \sigma^2)$$

Likelihood of $(\vec{w}, \sigma^2)$: Using fact that training examples are treated as IID,

$$L(\vec{w}, \sigma^2) \;=\; p_{\vec{w},\sigma^2}((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)) \;\propto\; \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \vec{x}_i \cdot \vec{w})^2}{2\sigma^2}\right)$$

(constant of proportionality doesn't depend on $\vec{w}$ or σ^2)

Log-likelihood of $(\vec{w}, \sigma^2)$:

$$\ln L(\vec{w}, \sigma^2) \;=\; -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \vec{x}_i \cdot \vec{w})^2 \;+\; \frac{n}{2} \ln \frac{1}{\sigma^2} \;+\; (\text{terms that don't depend on } \vec{w} \text{ or } \sigma^2)$$

Maximizing $\vec{w} \mapsto \ln L(\vec{w}, \sigma^2)$ is same minimizing SSE on training data

MLE for normal linear regression model

Suppose training data $\mathcal{S}$ are $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$

Recall: Under normal linear regression model with parameters $\vec{w}$ and σ^2,

$$(Y_i \mid \vec{X}_i = \vec{x}_i) \sim \mathrm{N}(\vec{x}_i \cdot \vec{w}, \sigma^2)$$

Likelihood of $(\vec{w}, \sigma^2)$: Using fact that training examples are treated as IID,

$$L(\vec{w}, \sigma^2) \;=\; p_{\vec{w},\sigma^2}((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)) \;\propto\; \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \vec{x}_i \cdot \vec{w})^2}{2\sigma^2}\right)$$

(constant of proportionality doesn't depend on $\vec{w}$ or σ^2)

Log-likelihood of $(\vec{w}, \sigma^2)$:

$$\ln L(\vec{w}, \sigma^2) \;=\; -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \vec{x}_i \cdot \vec{w})^2 \;+\; \frac{n}{2} \ln \frac{1}{\sigma^2} \;+\; (\text{terms that don't depend on } \vec{w} \text{ or } \sigma^2)$$

Maximizing $\vec{w} \mapsto \ln L(\vec{w}, \sigma^2)$ is same minimizing SSE on training data

Upshot: MLE for normal linear regression model parameter $\vec{w}$ is same as OLS

Plug-in principle

Applying the plug-in principle:

Plug-in principle

Applying the plug-in principle:

1. Assume normal linear regression model with parameters $\vec{w}$ and σ^2

Plug-in principle

Applying the plug-in principle:

1. Assume normal linear regression model with parameters $\vec{w}$ and σ^2
2. Optimal predictor (the regression function $\eta(\vec{x}) = \mathbb{E}[Y \mid \vec{X} = \vec{x}]$) is linear function

$$f^\star(\vec{x}) = \vec{x} \cdot \vec{w}$$

   which involves unknown parameter $\vec{w}$

Plug-in principle

Applying the plug-in principle:

1. Assume normal linear regression model with parameters $\vec{w}$ and σ^2
2. Optimal predictor (the regression function $\eta(\vec{x}) = \mathbb{E}[Y \mid \vec{X} = \vec{x}]$) is linear function

$$f^\star(\vec{x}) = \vec{x} \cdot \vec{w}$$

   which involves unknown parameter $\vec{w}$
3. Estimate $\vec{w}$ using training data (via MLE) $\rightarrow \vec{w}_{\mathrm{mle}}$; learned predictor is

$$f(\vec{x}) = \vec{x} \cdot \vec{w}_{\mathrm{mle}}$$

What about the conditional variance?

Can also estimate the σ^2 parameter in the normal linear regression model via MLE:

$$\sigma^2_{\text{mle}} := \frac{1}{n}\sum_{i=1}^{n}(\vec{x}_i \cdot \vec{w}_{\text{mle}} - y_i)^2$$

i.e., the MSE of $\vec{w}_{\text{mle}}$ on the training data

What about the conditional variance?

Can also estimate the σ^2 parameter in the normal linear regression model via MLE:

$$\sigma^2_{\text{mle}} := \frac{1}{n}\sum_{i=1}^{n}(\vec{x}_i \cdot \vec{w}_{\text{mle}} - y_i)^2$$

i.e., the MSE of $\vec{w}_{\text{mle}}$ on the training data

- In linear regression model with parameters $(\vec{w}, \sigma^2)$, conditional variance σ^2 *is* the MSE of $\vec{w}$:

  $$\mathbb{E}[(\vec{X} \cdot \vec{w} - Y)^2] = \sigma^2$$

  and σ^2_{mle} is a reasonable estimate of this quantity

What about the conditional variance?

Can also estimate the σ^2 parameter in the normal linear regression model via MLE:

$$\sigma^2_{\text{mle}} := \frac{1}{n}\sum_{i=1}^{n}(\vec{x}_i \cdot \vec{w}_{\text{mle}} - y_i)^2$$

i.e., the MSE of $\vec{w}_{\text{mle}}$ on the training data

- In linear regression model with parameters $(\vec{w}, \sigma^2)$, conditional variance σ^2 *is* the MSE of $\vec{w}$:

  $$\mathbb{E}[(\vec{X} \cdot \vec{w} - Y)^2] = \sigma^2$$

  and σ^2_{mle} is a reasonable estimate of this quantity
- **Caution:** σ^2_{mle} might be a bad estimate of how well $\vec{w}_{\text{mle}}$ predicts on new data, i.e., of

  $$\mathbb{E}[(\vec{X} \cdot \vec{w}_{\text{mle}} - Y)^2 \mid \vec{w}_{\text{mle}}]$$

  Typically a *biased* (under-)estimate
  (Recall: Both σ^2_{mle} and $\vec{w}_{\text{mle}}$ depend on training data!)

What about the conditional variance?

Can also estimate the σ^2 parameter in the normal linear regression model via MLE:

$$\sigma^2_{\text{mle}} := \frac{1}{n}\sum_{i=1}^{n}(\vec{x}_i \cdot \vec{w}_{\text{mle}} - y_i)^2$$

i.e., the MSE of $\vec{w}_{\text{mle}}$ on the training data

- In linear regression model with parameters $(\vec{w}, \sigma^2)$, conditional variance σ^2 *is* the MSE of $\vec{w}$:

  $$\mathbb{E}[(\vec{X} \cdot \vec{w} - Y)^2] = \sigma^2$$

  and σ^2_{mle} is a reasonable estimate of this quantity
- **Caution:** σ^2_{mle} might be a bad estimate of how well $\vec{w}_{\text{mle}}$ predicts on new data, i.e., of

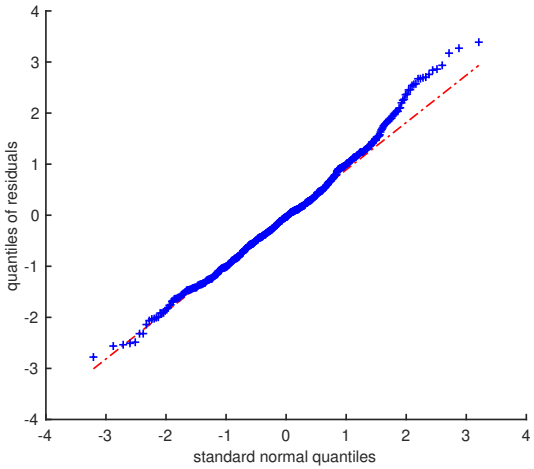
  $$\mathbb{E}[(\vec{X} \cdot \vec{w}_{\text{mle}} - Y)^2 \mid \vec{w}_{\text{mle}}]$$

  Typically a *biased* (under-)estimate
  (Recall: Both σ^2_{mle} and $\vec{w}_{\text{mle}}$ depend on training data!)
  - Instead, use test data (as usual)

Regression diagnostics

Diagnostics help assess suitability of the normal linear regression model

- correlations of residuals and features
- distribution of residuals
- ...



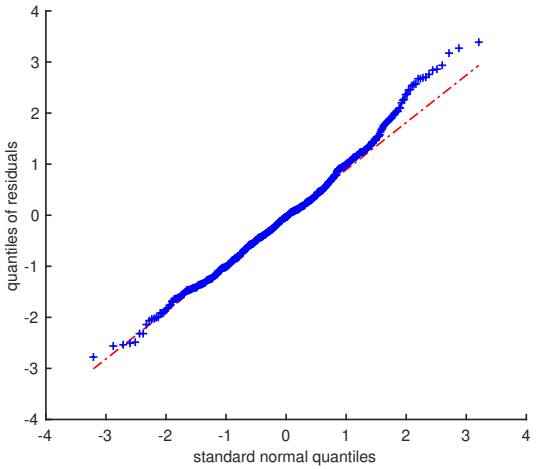
Regression diagnostics

Diagnostics help assess suitability of the normal linear regression model

- correlations of residuals and features
- distribution of residuals
- ...

Caution: These do not necessarily assess how well $\vec{w}_{\mathrm{mle}}$ will predict on new data

- For that, use test data (as usual)



Epilogue and recap

- In normal linear regression model, optimal predictor is a linear function
- OLS = MLE in this model
  - **Caution:** Theory about MLE is primarily developed under "large sample asymptotics"
  - In particular, theory suggests n should be (much) larger than d for a good estimate
- OLS (for supervised learning) is an instance of applying the plug-in principle
- Other inferences / diagnostics are justified by normal linear regression model
  - They do not eliminate need for test data to evaluate learned predictor

Statistical Learning View of OLS

Motivation

Statistical learning view of OLS:

- Can we justify OLS without the normality assumptions?
- Broader perspective on supervised learning

Relaxing the normal linear regression model

IID model for training data:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID

Relaxing the normal linear regression model

IID model for training data:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID

Normal linear regression model:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID, and for any $\vec{x} \in \mathbb{R}^d$,

$$(Y \mid \vec{X} = \vec{x}) \sim \mathrm{N}(\vec{x} \cdot \vec{w}, \sigma^2)$$

Relaxing the normal linear regression model

IID model for training data:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID

IID model where there is a "good" linear function ...
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID, and there exists $\vec{w} \in \mathbb{R}^d$ such that

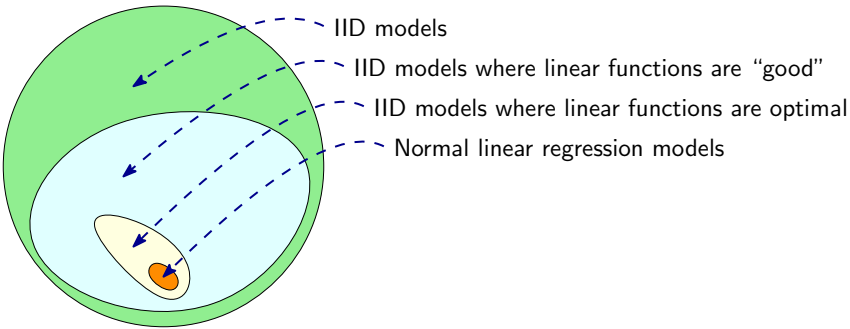
$$\mathbb{E}[(\vec{X} \cdot \vec{w} - Y)^2] \quad \text{is "relatively small"}$$

Normal linear regression model:
Training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$ and "future" example $(\vec{X}, Y)$ are IID, and for any $\vec{x} \in \mathbb{R}^d$,

$$(Y \mid \vec{X} = \vec{x}) \sim \mathrm{N}(\vec{x} \cdot \vec{w}, \sigma^2)$$

Relating the models

Relationship between models for regression:



Empirical distribution

Let P denote the distribution of (X, Y) in the IID model

- If P was "known", then we could find linear function with smallest MSE (in terms of P)

Empirical distribution

Let P denote the distribution of (X, Y) in the IID model

- If P was "known", then we could find linear function with smallest MSE (in terms of P)

Of course, P is typically unknown, but perhaps we can estimate it (à la plug-in principle)!

Empirical distribution

Let P denote the distribution of (X, Y) in the IID model

- If P was "known", then we could find linear function with smallest MSE (in terms of P)

Of course, P is typically unknown, but perhaps we can estimate it (à la plug-in principle)!

Generic estimate of P based on training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$: **empirical distribution** P_n

- To draw a random example from P_n:
  - Pick a number i uniformly at random from $\{1, \ldots, n\}$
  - Return $(\vec{X}_i, Y_i)$

Empirical distribution

Let P denote the distribution of (X, Y) in the IID model

- If P was "known", then we could find linear function with smallest MSE (in terms of P)

Of course, P is typically unknown, but perhaps we can estimate it (à la plug-in principle)!

Generic estimate of P based on training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$: **empirical distribution** P_n

- To draw a random example from P_n:
  - Pick a number i uniformly at random from $\{1, \ldots, n\}$
  - Return $(\vec{X}_i, Y_i)$

Upshot: Taking an expectation with respect to P_n $\equiv$ Taking a sample average over training data

Empirical distribution

Let P denote the distribution of (X, Y) in the IID model

- If P was "known", then we could find linear function with smallest MSE (in terms of P)

Of course, P is typically unknown, but perhaps we can estimate it (à la plug-in principle)!

Generic estimate of P based on training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$: **empirical distribution** P_n

- To draw a random example from P_n:
  - Pick a number i uniformly at random from $\{1, \ldots, n\}$
  - Return $(\vec{X}_i, Y_i)$

Upshot: Taking an expectation with respect to P_n $\equiv$ Taking a sample average over training data

Theorem (informal). Under IID model for training data when n is "large", P_n is "typically" "close to" P, at least in ways that matter to ML algorithms for "simple predictors"

Empirical distribution

Let P denote the distribution of (X, Y) in the IID model

- If P was "known", then we could find linear function with smallest MSE (in terms of P)

Of course, P is typically unknown, but perhaps we can estimate it (à la plug-in principle)!

Generic estimate of P based on training data $(\vec{X}_1, Y_1), \ldots, (\vec{X}_n, Y_n)$: **empirical distribution** P_n

- To draw a random example from P_n:
  - Pick a number i uniformly at random from $\{1, \ldots, n\}$
  - Return $(\vec{X}_i, Y_i)$

Upshot: Taking an expectation with respect to P_n $\equiv$ Taking a sample average over training data

> **Theorem (informal).** Under IID model for training data when n is "large", P_n is "typically" "close to" P, at least in ways that matter to ML algorithms for "simple predictors"

Theorem doesn't necessarily apply to ML algorithms that can return "arbitrarily complex predictors" (e.g., decision trees of arbitrary size)

Empirical risk minimization (for linear regression)

- MSE with respect to P: **population risk**

$$\mathbb{E}[(f(\vec{X}) - Y)^2]$$

where $(\vec{X}, Y) \sim P$

Empirical risk minimization (for linear regression)

- MSE with respect to P: **population risk**

$$\mathbb{E}[(f(\vec{X}) - Y)^2]$$

where $(\vec{X}, Y) \sim P$

- MSE with respect to P_n (i.e., MSE on training data): **empirical risk**

$$\frac{1}{n}\sum_{i=1}^{n}(f(\vec{X}_i) - Y_i)^2$$

(When drawing random example from P_n, each $(\vec{X}_i, Y_i)$ has probability $1/n$ of being picked)

Empirical risk minimization (for linear regression)

- MSE with respect to P: **population risk**

$$\mathbb{E}[(f(\vec{X}) - Y)^2]$$

  where $(\vec{X}, Y) \sim P$

- MSE with respect to P_n (i.e., MSE on training data): **empirical risk**

$$\frac{1}{n}\sum_{i=1}^{n}(f(\vec{X}_i) - Y_i)^2$$

  (When drawing random example from P_n, each $(\vec{X}_i, Y_i)$ has probability $1/n$ of being picked)

- OLS finds $\vec{w}$ that minimizes empirical risk among all linear functions
  - I.e., OLS performs **empirical risk minimization (ERM)** for linear functions

Empirical risk minimization (for linear regression)

- MSE with respect to P: **population risk**

$$\mathbb{E}[(f(\vec{X}) - Y)^2]$$

  where $(\vec{X}, Y) \sim P$

- MSE with respect to P_n (i.e., MSE on training data): **empirical risk**

$$\frac{1}{n}\sum_{i=1}^{n}(f(\vec{X}_i) - Y_i)^2$$

  (When drawing random example from P_n, each $(\vec{X}_i, Y_i)$ has probability $1/n$ of being picked)

- OLS finds $\vec{w}$ that minimizes empirical risk among all linear functions
  - I.e., OLS performs **empirical risk minimization (ERM)** for linear functions

Theorem (informal). Under IID model for n training data, ERM linear function $\vec{w}_{\text{erm}}$ "typically" satisfies

$$[\text{Population risk of } \vec{w}_{\text{erm}}] \leq \min_{\vec{w} \in \mathbb{R}^d} [\text{Population risk of } \vec{w}] + [\text{a small number if } n \gg d]$$

Epilogue and recap

- Statistical learning view of OLS provides justification under broader modeling assumptions
- OLS = ERM in this model
  - **Caution:** Theory about ERM is a bit murky
  - Theory suggests n should (much) larger than d for good ERM performance
- ERM is an instance of applying the plug-in principle