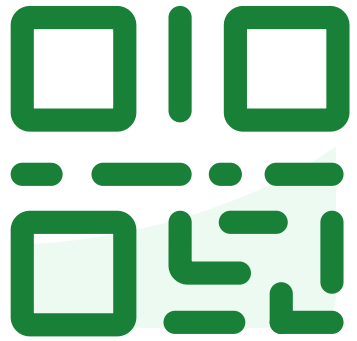


Do not edit
How to change the
design



**Join at slido.com
#2060756**

① The Slido app must be installed on every computer you're presenting from

slido

Lecture 6

Linear Regression (1)

Model, error function, normal equations, geometry, evaluation, and regularization

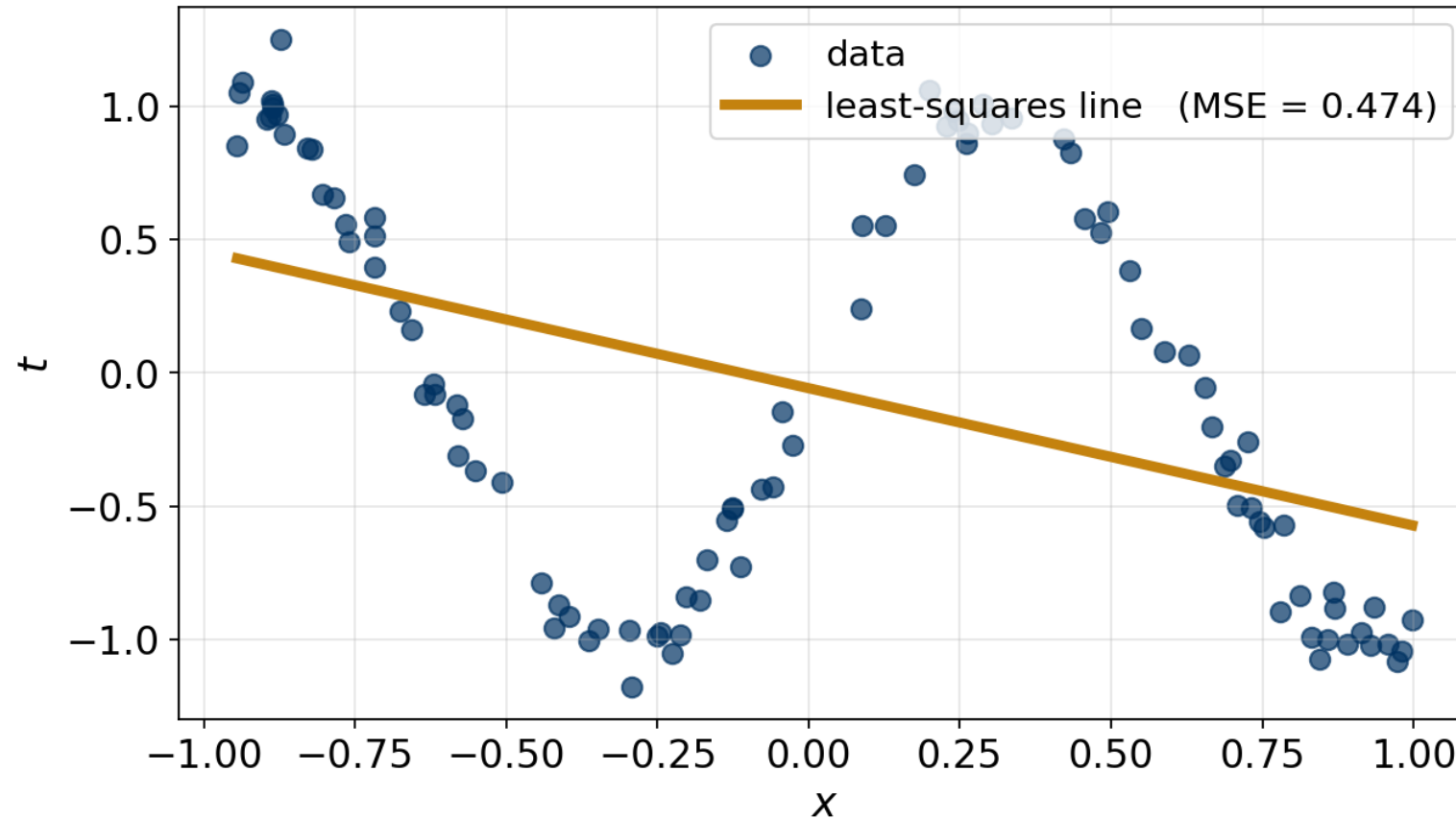
EECS 189/289, Fall 2026 @ UC Berkeley

Joseph E. Gonzalez and Narges Norouzi

Fitting a Straight Line to Non-Linear Data



2060756



Roadmap

- Linear Regression
- Basis Functions
- Vectorizing Calculations
- Error Function
- Error Function Minimization
- Geometric Interpretation
- Evaluation



2060756

Linear Regression

- Linear Regression
- Basis Functions
- Vectorizing Calculations
- Error Function
- Error Function Minimization
- Geometric Interpretation
- Evaluation

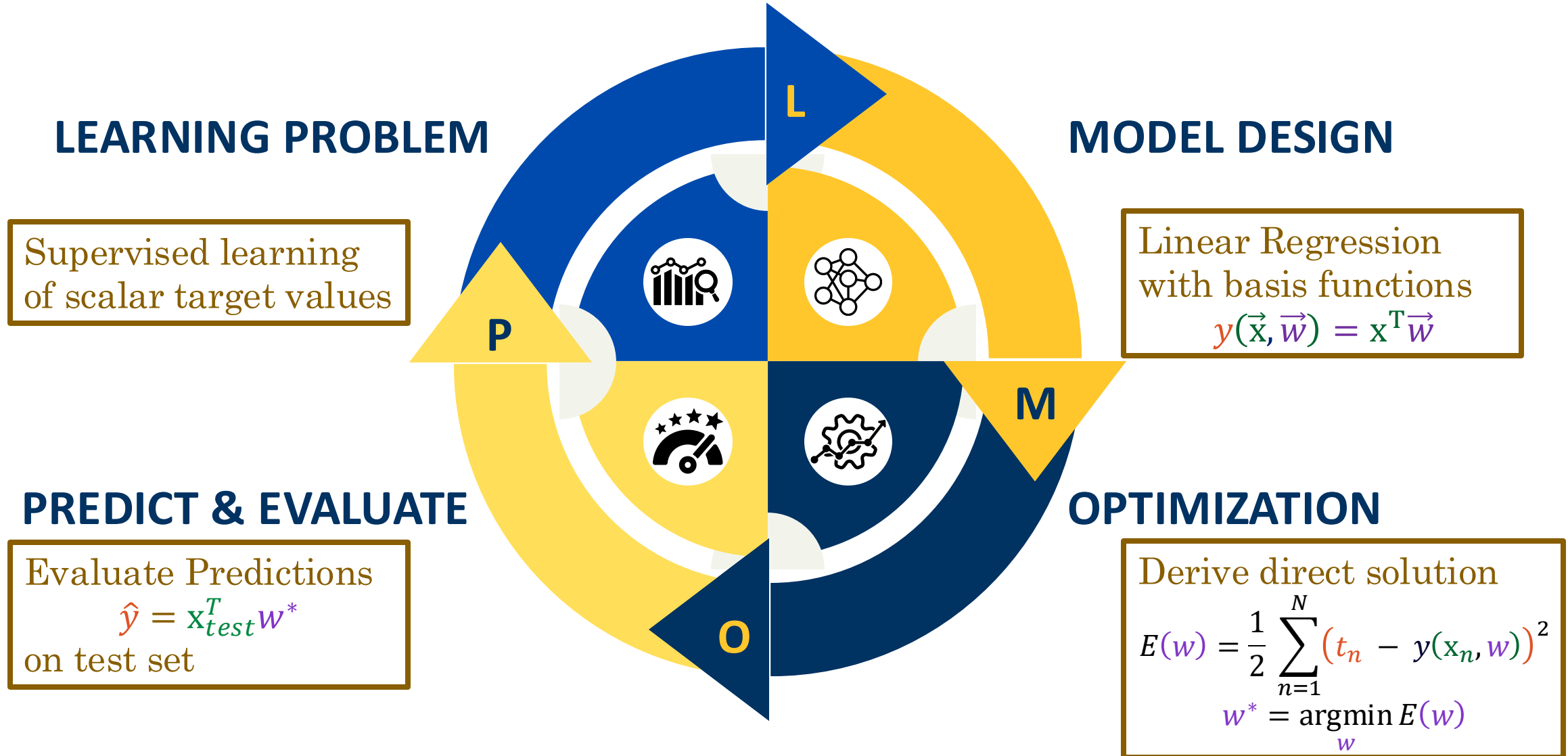


2060756

Linear Regression Outline



2060756



Learning Problem



2060756

LEARNING PROBLEM

Supervised learning
of scalar target values

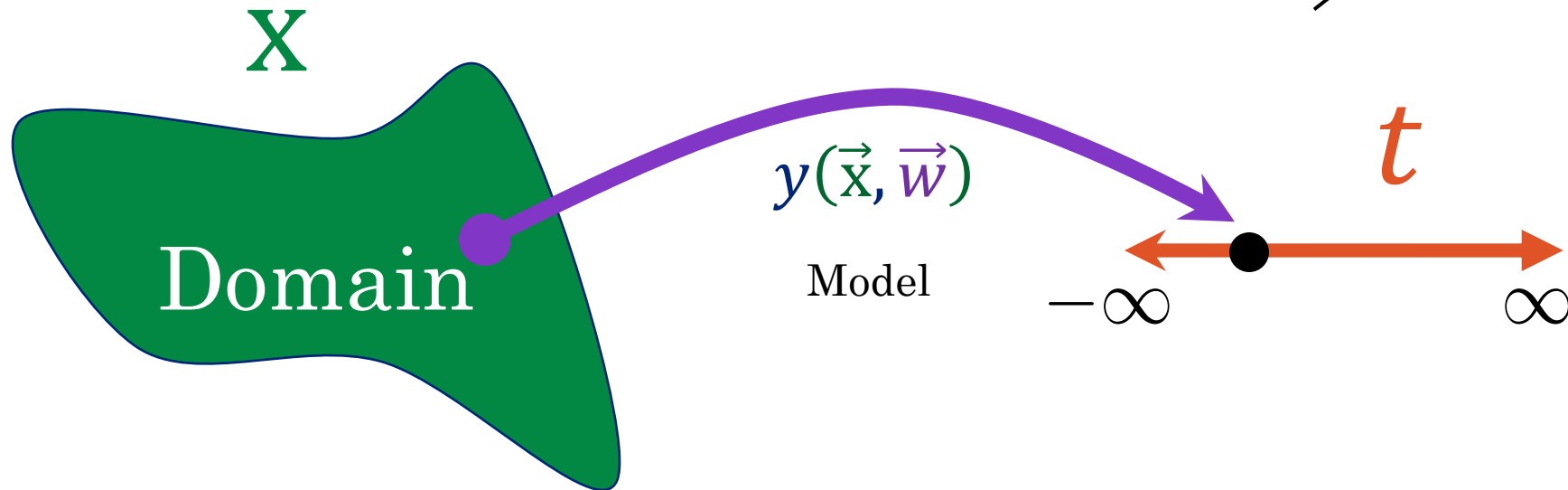
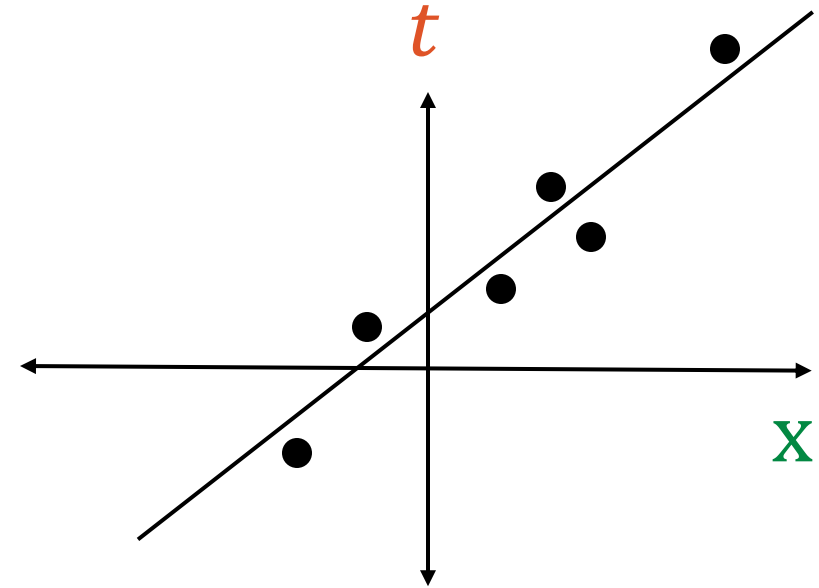


Regression



2060756

- Estimating relationship between $\mathbf{x}$ and t .
 - t is a quantitative value
 - We will soon see $\mathbf{x}$ can be almost anything ...



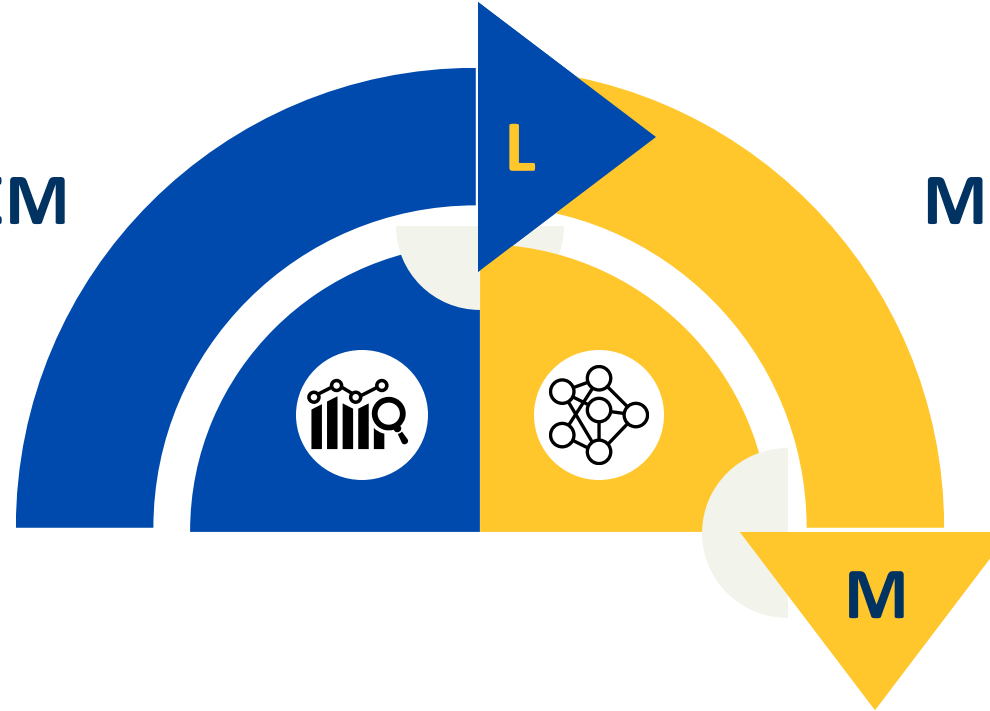
Model Design



2060756

LEARNING PROBLEM

Supervised learning
of scalar target values



MODEL DESIGN

Linear Regression
with basis functions
 $y(\vec{x}, \vec{w}) = \vec{x}^T \vec{w}$



2060756

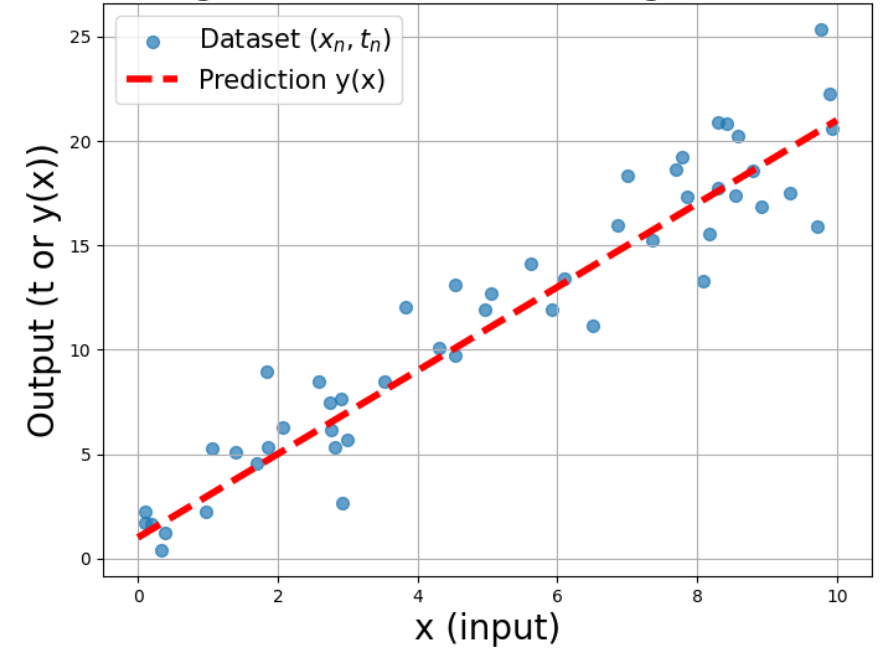
Supervised Linear Regression

- Goal: Predict one or more continuous *target* variable t given the vector $\vec{x} \in \mathbb{R}^D$ of *input* variables.

$$\{(x_n)\}_{n=1}^N \xrightarrow{\vec{w}} \{(t_n)\}_{n=1}^N$$

- We formulate a function $y(\vec{x}, \vec{w})$ whose values for new input $\vec{x}$ constitute the predictions for the corresponding values of t .
- $\vec{w}$ is a vector of parameters that can be learned from the training data.

Visualizing Dataset and a Linear Regression Function



The Simplest Linear Regression Model



2060756

- The simplest model is of the following form:

$$y(\vec{x}, \vec{w}) = w_0 + w_1 x_1 + \dots + w_D x_D \quad \text{where } \vec{x} = (x_1, x_2, \dots, x_D)^T \in \mathbb{R}^D.$$

Predicted output $\downarrow$

Linear combination of parameters $w_0, w_1, \dots, w_D$, each scaled by x_d 's.

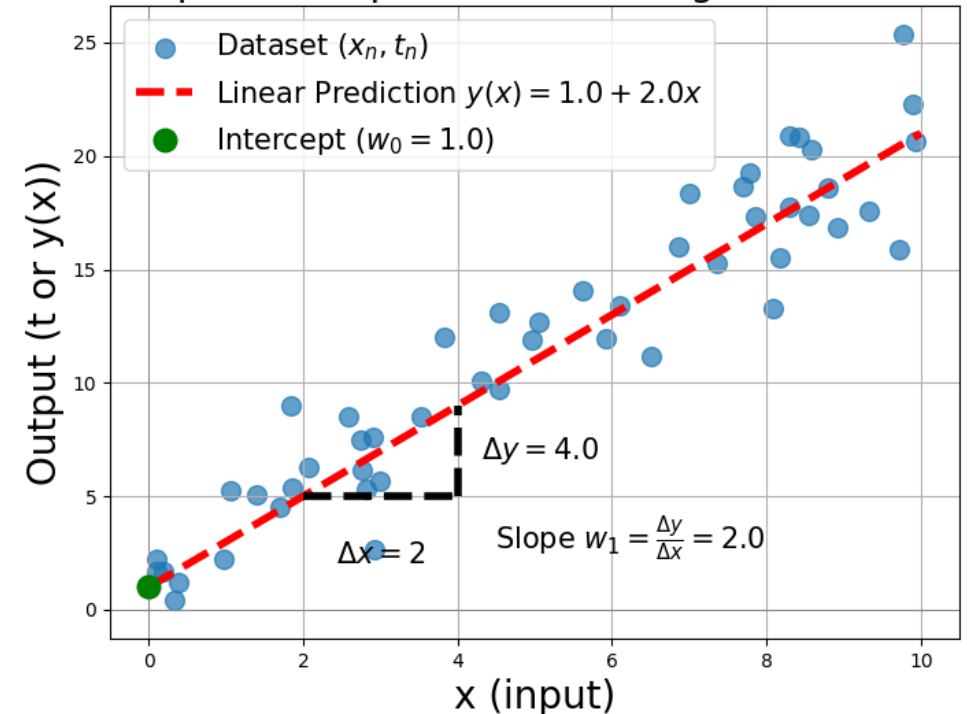
- In the simplest case of $D = 1$, we have:

$$y(\vec{x}, \vec{w}) = w_0 + w_1 x_1$$

Intercept (shift)

Slope (rate of change)

Intercept and Slope in a Linear Regression Function





2060756

The Simplest Linear Regression Model

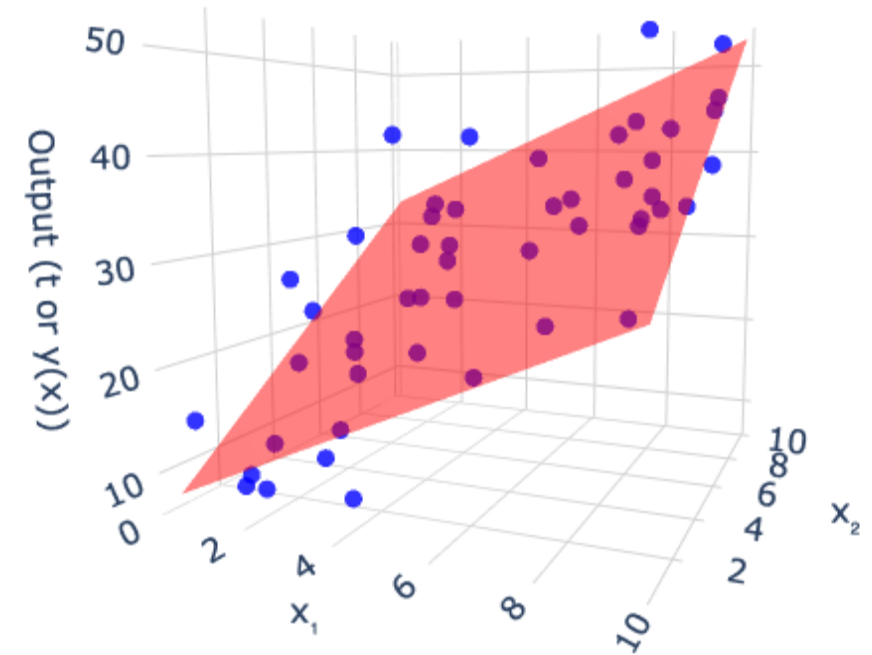
- The simplest model is of the following form:

$$y(\vec{x}, \vec{w}) = w_0 + w_1 x_1 + \dots + w_D x_D \quad \text{where } \vec{x} = (x_1, x_2, \dots, x_D)^T \in \mathbb{R}^D.$$

$y(\vec{x}, \vec{w})$ ↓ Predicted output

$w_0 + w_1 x_1 + \dots + w_D x_D$ Linear combination of parameters $w_0, w_1, \dots, w_D$, each scaled by x_d 's.

- In higher dimension, the predicted values form a hyperplane in $\mathbb{R}^{D+1}$.



[Interactive link](#)



Which of the following is a linear regression model?



2060756

Basis Functions

- Linear Regression
- Basis Functions
- Vectorizing Calculations
- Error Function
- Error Function Minimization
- Geometric Interpretation
- Evaluation

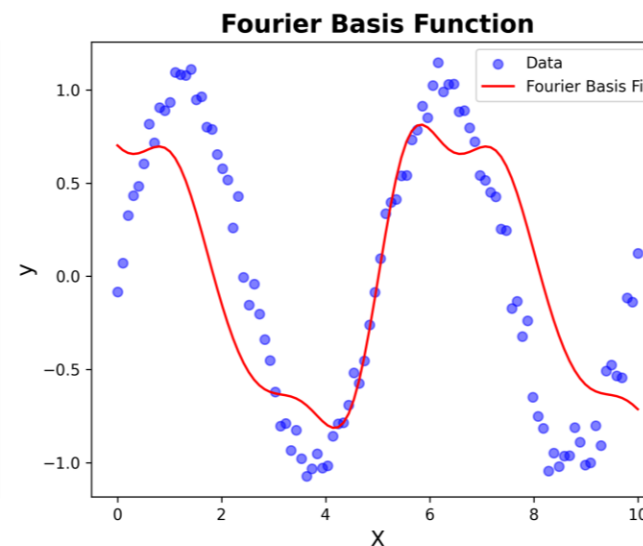
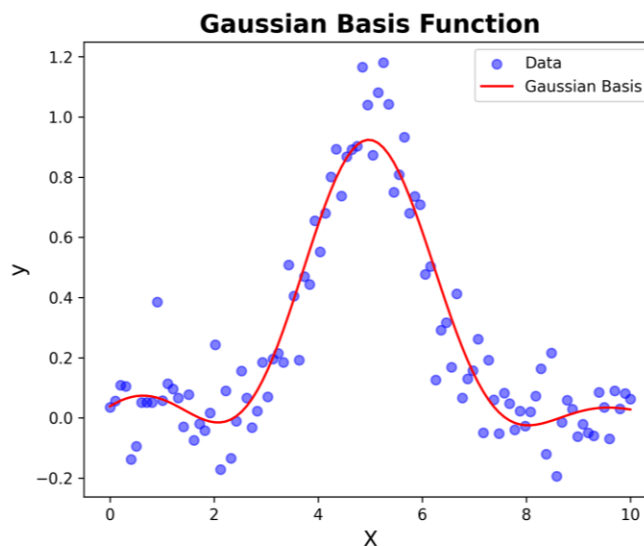
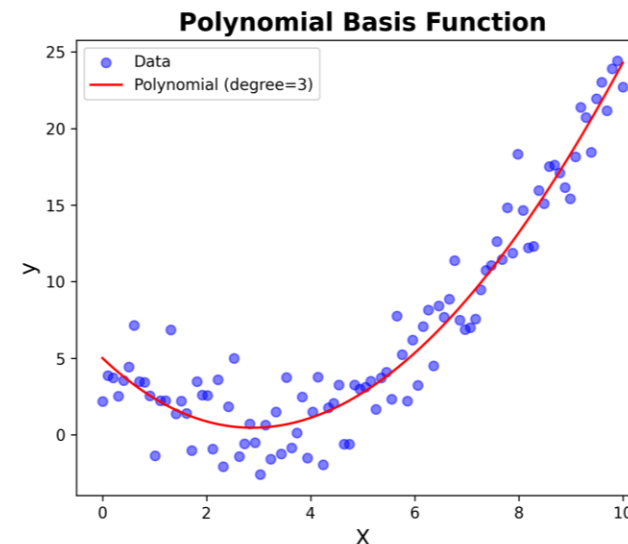
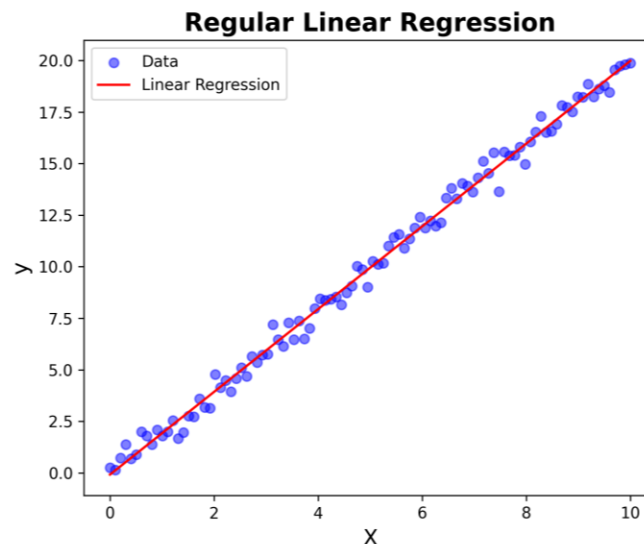
Linear Functions From Slido

These are all linear models with different basis functions.

We will now see what basis functions are...



2060756





What Does It Mean To Be a Linear Model?

In what sense are the previous plots linearly modeled?

$$y(\vec{x}, \vec{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

Are linear models linear in the

1. Features?
2. Parameters?



Are linear models linear in the ...



2060756

What Does It Mean To Be a Linear Model?

In what sense are the previous plots linearly modeled?

$$y(\vec{x}, \vec{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

Linear in the Parameters

Feature Functions

Are linear models linear in the

1. Features?
2. Parameters?



2060756

Basis Functions

- We can extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables, of the form:

$$y(\vec{x}, \vec{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

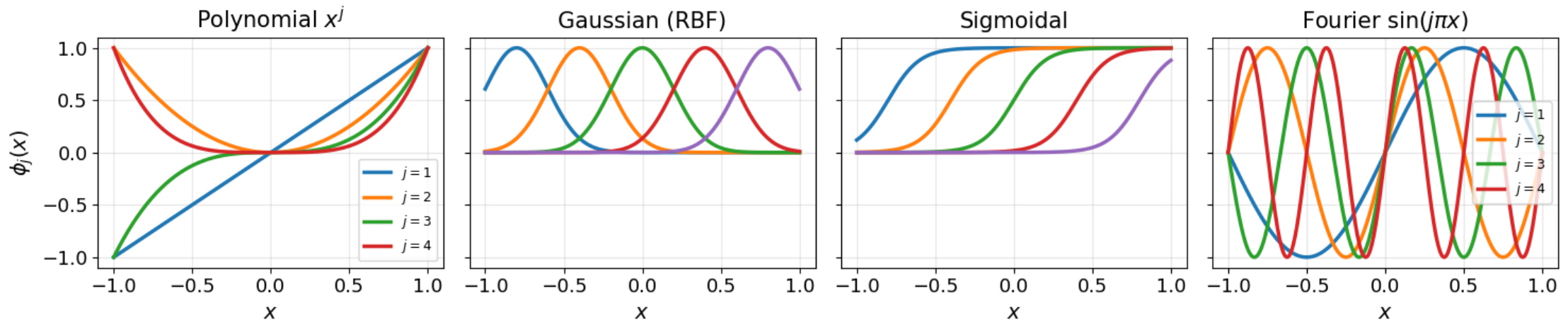
where $\phi_j(\mathbf{x})$ are known as basis functions.



2060756

More on Basis Function

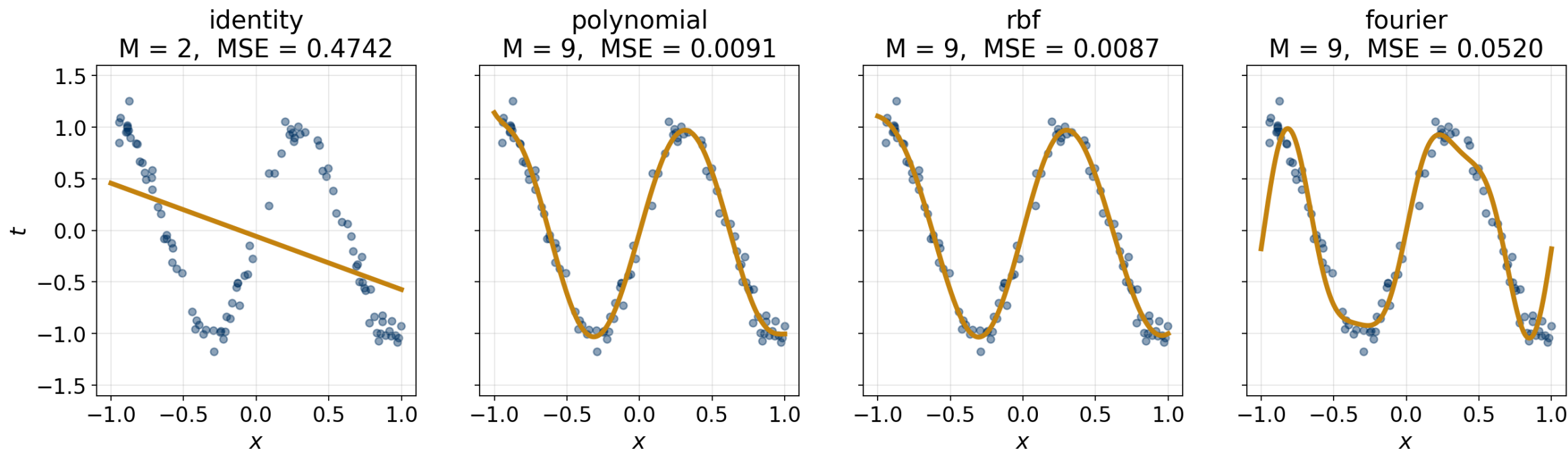
Family	$\phi_j(x)$	Behaviour
Polynomial	x^j	global — changing one w_j moves the whole curve
Gaussian (RBF)	$\exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$	local — each w_j affects a neighbourhood of μ_j
Sigmoidal	$\sigma\left(\frac{x-\mu_j}{s}\right), \sigma(a) = \frac{1}{1+e^{-a}}$	local step at μ_j
Fourier	$\sin(jx), \cos(jx)$	periodic



Comparing Basis Function Families



2060756



The same least-squares fit applied four times. Only ϕ_j differs.

MSE: identity 0.474, polynomial 0.0091, RBF 0.0087, Fourier 0.0520.

Vectorizing Calculations

- Linear Regression
- Basis Functions
- Vectorizing Calculations
- Error Function
- Error Function Minimization
- Geometric Interpretation
- Evaluation



2060756



2060756

Vectorizing Calculations

- Rewriting the equation for linear regression:

$$y(\vec{x}, \vec{w}) = w_0 + w_1 x_1 + \dots + w_D x_D = w_0 + \sum_{d=1}^D w_d x_d$$

- If we assume that there is an $x_0 = 1$, then the two terms above can be combined into one summation:

$$y(\vec{x}, \vec{w}) = w_0 + w_1 x_1 + \dots + w_D x_D = \sum_{d=0}^D w_d x_d$$

where $\vec{x} = (x_0, x_1, x_2, \dots, x_D)^T$ and $\vec{w} = (w_0, w_1, w_2, \dots, w_D)^T$

↓
Augmented input with x_0

↘
 $x_0 = 1$



2060756

Vectorizing Calculations

$$y(\vec{x}, \vec{w}) = w_0 + w_1 x_1 + \dots + w_D x_D = \sum_{d=0}^D w_d x_d$$

where $\vec{x} = (x_0, x_1, x_2, \dots, x_D)^T$ and $\vec{w} = (w_0, w_1, w_2, \dots, w_D)^T$

$$\begin{aligned} y(\vec{x}, \vec{w}) &= \vec{w} \cdot \vec{x} \\ &= \vec{w}^T \vec{x} = \vec{x}^T \vec{w} \end{aligned}$$



2060756

Matrix Notation

- Assuming we have $n = 1, \dots, N$ samples in our dataset, we can write:


$$\left\{ \begin{array}{l} y(\vec{x}_1, \vec{w}) = w_0 + w_1 x_{11} + w_2 x_{12} + \dots + w_D x_{1D} \\ y(\vec{x}_2, \vec{w}) = w_0 + w_1 x_{21} + w_2 x_{22} + \dots + w_D x_{2D} \\ \dots \\ y(\vec{x}_N, \vec{w}) = w_0 + w_1 x_{N1} + w_2 x_{N2} + \dots + w_D x_{ND} \end{array} \right.$$
$$\left\{ \begin{array}{ll} y(\vec{x}_1, \vec{w}) = \vec{x}_1^T \vec{w} & \text{where } \vec{x}_1^T = [1, x_{11}, x_{12}, \dots, x_{1D}] \\ y(\vec{x}_2, \vec{w}) = \vec{x}_2^T \vec{w} & \text{where } \vec{x}_2^T = [1, x_{21}, x_{22}, \dots, x_{2D}] \\ \dots & \dots \\ y(\vec{x}_N, \vec{w}) = \vec{x}_N^T \vec{w} & \text{where } \vec{x}_N^T = [1, x_{N1}, x_{N2}, \dots, x_{ND}] \end{array} \right.$$

Matrix Notation



2060756

$$\begin{cases} y(\vec{x}_1, \vec{w}) = \vec{x}_1^T \vec{w} & \text{where } \vec{x}_1^T = [1, x_{11}, x_{12}, \dots, x_{1D}] \\ y(\vec{x}_2, \vec{w}) = \vec{x}_2^T \vec{w} & \text{where } \vec{x}_2^T = [1, x_{21}, x_{22}, \dots, x_{2D}] \\ \dots \\ y(\vec{x}_N, \vec{w}) = \vec{x}_N^T \vec{w} & \text{where } \vec{x}_N^T = [1, x_{N1}, x_{N2}, \dots, x_{ND}] \end{cases}$$

$$\begin{cases} y(\vec{x}_1, \vec{w}) = [1, x_{11}, x_{12}, \dots, x_{1D}] \vec{w} \\ y(\vec{x}_2, \vec{w}) = [1, x_{21}, x_{22}, \dots, x_{2D}] \vec{w} \\ \dots \\ y(\vec{x}_N, \vec{w}) = [1, x_{N1}, x_{N2}, \dots, x_{ND}] \vec{w} \end{cases}$$

$$\mathbb{X} = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\} \in \mathbb{R}^{N \times (D+1)}$$

Design matrix

$$\mathbb{Y} = \{y(\vec{x}_1, \vec{w}), y(\vec{x}_2, \vec{w}), \dots, y(\vec{x}_N, \vec{w})\} \in \mathbb{R}^N$$

$$\mathbb{Y}(\mathbb{X}, \vec{w}) = \mathbb{X} \vec{w}$$

$$\in \mathbb{R}^{N \times (D+1)}$$

$$\in \mathbb{R}^{D+1}$$

Error Function

- Linear Regression
- Basis Functions
- Vectorizing Calculations
- Error Function
- Error Function Minimization
- Geometric Interpretation
- Evaluation



2060756

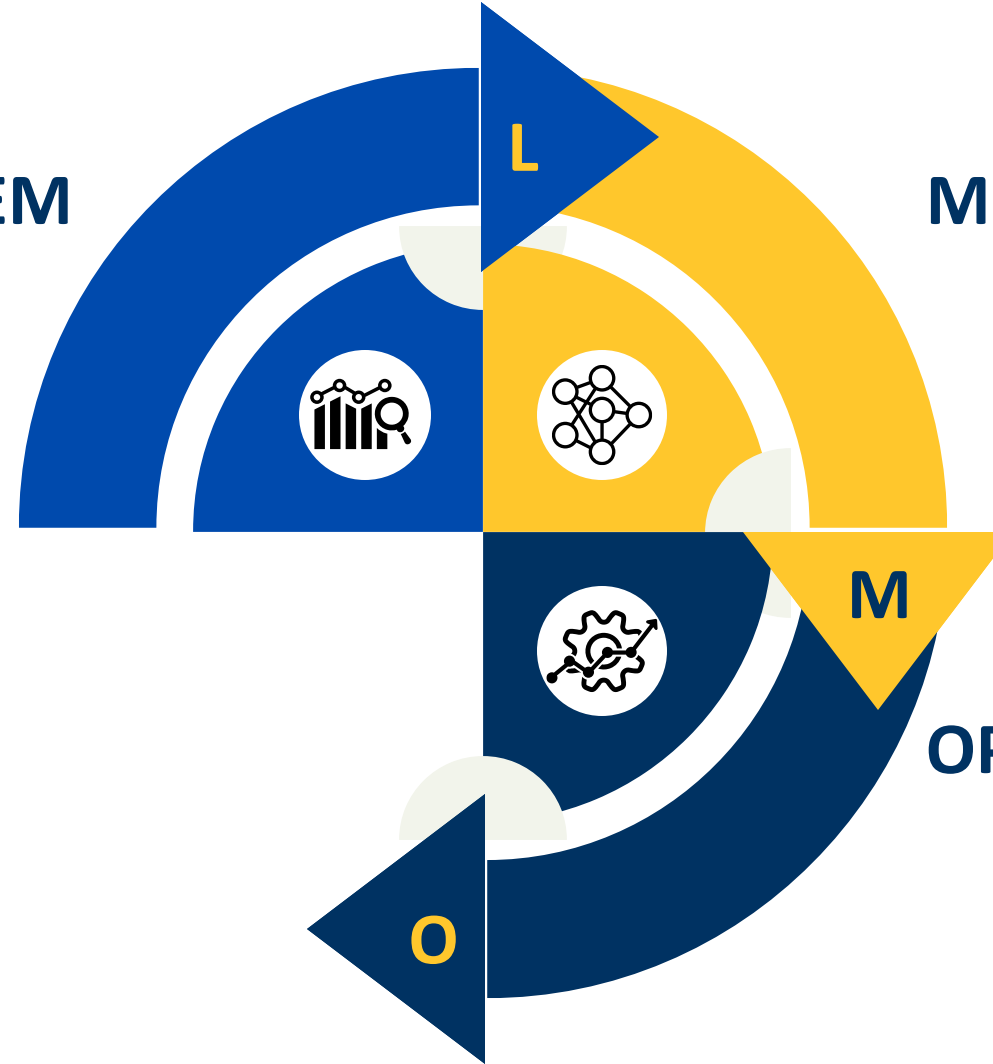
Optimization



2060756

LEARNING PROBLEM

Supervised learning
of scalar target values



MODEL DESIGN

Linear Regression
with basis functions
 $y(\vec{x}, \vec{w}) = \vec{x}^T \vec{w}$

OPTIMIZATION

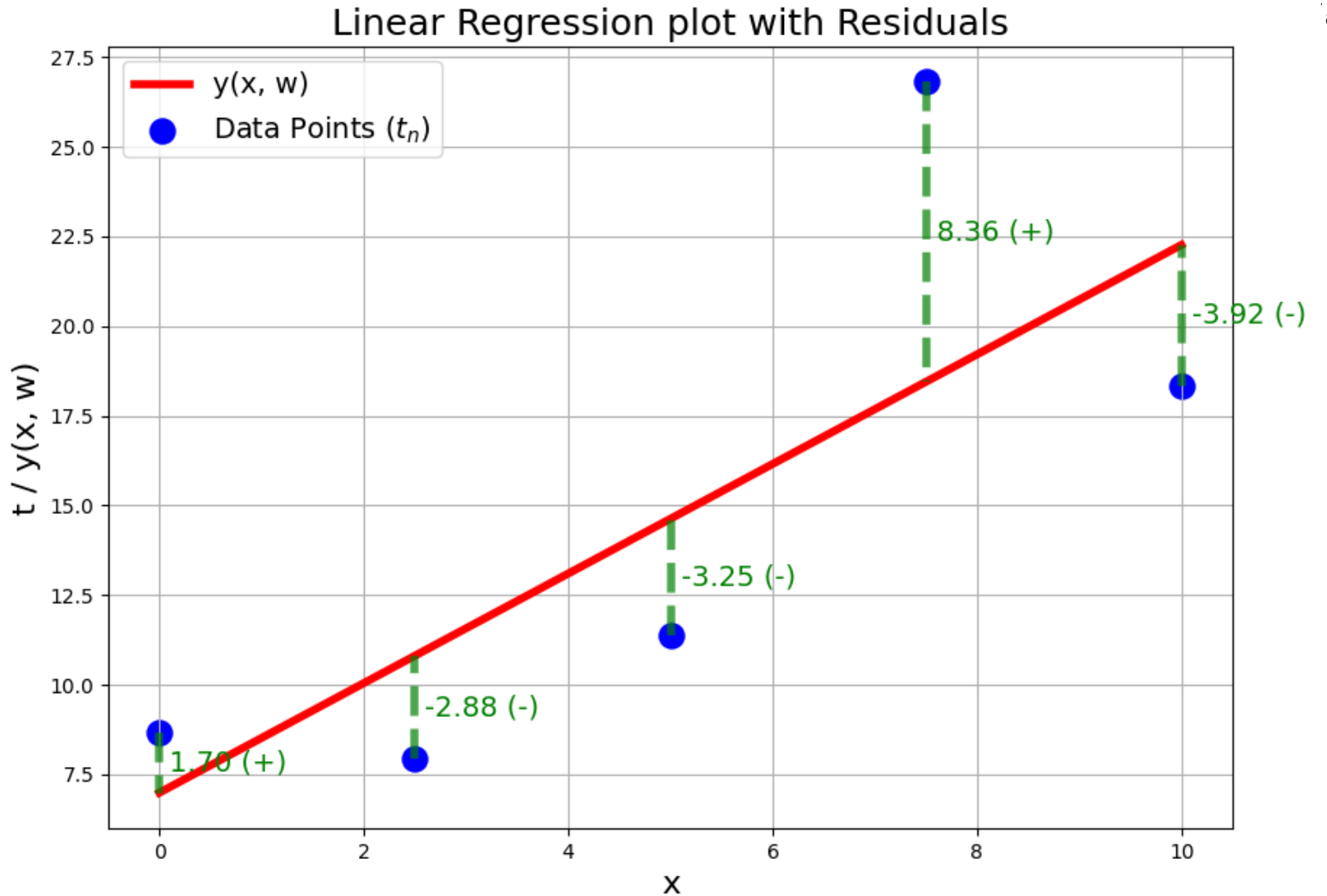
Derive direct solution

$$E(w) = \frac{1}{2} \sum_{n=1}^N (t_n - y(x_n, w))^2$$
$$w^* = \underset{w}{\operatorname{argmin}} E(w)$$

Error Function Visualization

Residuals are defined as
 $e_n = (t_n - y(x_n, w))$
and they can be + or -.

The error function $E(w)$
combines the squares of
the residuals to measure
the goodness of the fit.





2060756

Error Function

- How to calculate the weights $\vec{w}$?
- We do this by minimizing an **error function** that measures the **misfit** between the function $\mathbb{Y}(\mathbb{X}, \vec{w})$ and **targets**, for any given value of $\vec{w}$, and the training data $\mathbb{X}$.
- One simple error function: **Sum of the squares of the differences between targets and the predictions.**

$$E(w) = \frac{1}{2} \sum_{n=1}^N (t_n - y(x_n, w))^2$$

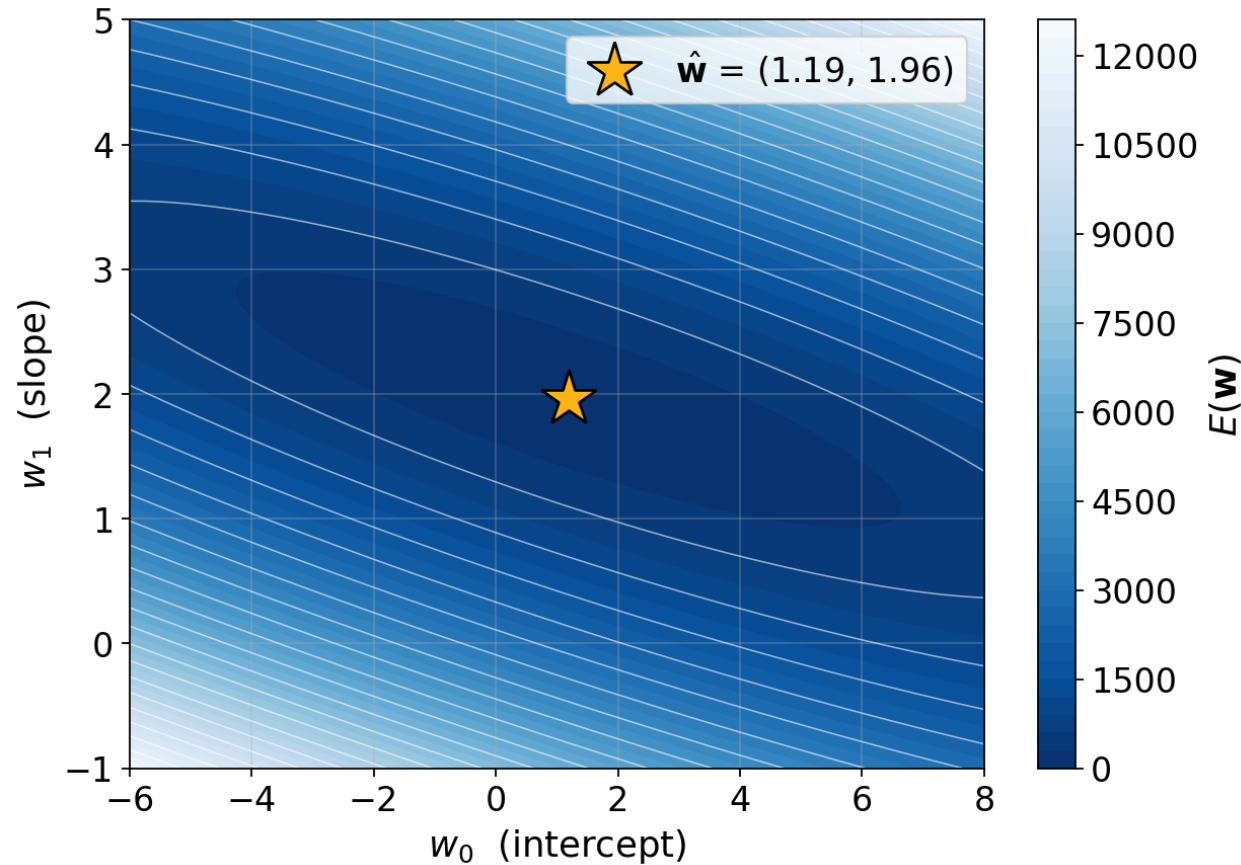
- Non-negative quantity.
- Only 0 if all predictions are equal to targets.

The factor of $\frac{1}{2}$ is included for later convenience.

Error Function Visualization



2060756



$E(\mathbf{w})$ for the two-parameter model.

The surface is convex with a single minimum.

Error Function Minimization

- Linear Regression
- Basis Functions
- Vectorizing Calculations
- Error Function
- Error Function Minimization
- Geometric Interpretation
- Evaluation



2060756



2060756

Error Function Minimization

- $E(w)$ is a quadratic function.

$$E(w) = \frac{1}{2} \sum_{n=1}^N (t_n - y(x_n, w))^2$$

Therefore, $\frac{\partial E(w)}{\partial w}$ is a linear function of w and hence solving $\frac{\partial E(w)}{\partial w} = 0$ has one and only one solution. Let's now derive this solution:

$$E(w) = \frac{1}{2} \sum_{n=1}^N \left(t_n - w_0 - \sum_{j=1}^D w_j x_{nj} \right)^2 \quad \xrightarrow{\text{Finding the optimum solution}} \quad \frac{\partial E(w)}{\partial w} = 0$$



2060756

Error Function Minimization

$$\frac{\partial E(w)}{\partial w_0} = 0 \longrightarrow \frac{\partial \left(\frac{1}{2} \sum_{n=1}^N \left((t_n - w_0 - \sum_{d=1}^D w_d x_{nd}) \right)^2 \right)}{\partial w_0} = 0$$

Sum rule

$$\partial \sum f = \sum \partial f$$

Chain rule

$$\partial f^2 = 2f \partial f$$

Separating
the terms

$$\frac{\partial E(w)}{\partial w_0} = 0 \longrightarrow$$



2060756

Error Function Minimization

$$\frac{\partial E(w)}{\partial w_0} = 0 \longrightarrow \frac{\partial \left(\frac{1}{2} \sum_{n=1}^N \left((t_n - w_0 - \sum_{d=1}^D w_d x_{nd}) \right)^2 \right)}{\partial w_0} = 0$$

Sum rule

$$\frac{\partial \sum f}{\partial w_0} = \sum \frac{\partial f}{\partial w_0} = \frac{1}{2} \sum_{n=1}^N \frac{\partial \left((t_n - w_0 - \sum_{d=1}^D w_d x_{nd})^2 \right)}{\partial w_0}$$

Chain rule
 $\frac{\partial f^2}{\partial f} = 2f \frac{\partial f}{\partial w_0}$

$$= \frac{1}{2} \sum_{n=1}^N 2 \left(t_n - w_0 - \sum_{d=1}^D w_d x_{nd} \right) \left(\frac{\partial (t_n - w_0 - \sum_{d=1}^D w_d x_{nd})}{\partial w_0} \right)$$

Separating
the terms

$$= \sum_{n=1}^N \left(t_n - w_0 - \sum_{d=1}^D w_d x_{nd} \right) (-1) = - \left(\sum_{n=1}^N t_n - n w_0 - \sum_{n=1}^N \left(\sum_{d=1}^D w_d x_{nd} \right) \right)$$

$$\frac{\partial E(w)}{\partial w_0} = 0 \longrightarrow w_0 = \frac{1}{n} \left(\sum_{n=1}^N t_n - \sum_{n=1}^N \left(\sum_{d=1}^D w_d x_{nd} \right) \right)$$



2060756

Error Function Minimization

$$\frac{\partial E(w)}{\partial w_k} = 0 \quad \longrightarrow \quad \frac{\partial \left(\frac{1}{2} \sum_{n=1}^N \left((t_n - w_0 - \sum_{d=1}^D w_d x_{nd}) \right)^2 \right)}{\partial w_k} = 0$$

Sum rule

$$\frac{\partial \sum f}{\partial w_k} = \sum \frac{\partial f}{\partial w_k} = \frac{1}{2} \sum_{n=1}^N \frac{\partial \left((t_n - w_0 - \sum_{d=1}^D w_d x_{nd})^2 \right)}{\partial w_k}$$

Chain rule

$$\frac{\partial f^2}{\partial f} = 2f \frac{\partial f}{\partial w_k} \quad = \frac{1}{2} \sum_{n=1}^N 2 \left(t_n - w_0 - \sum_{d=1}^D w_d x_{nd} \right) \left(\frac{\partial (t_n - w_0 - \sum_{d=1}^D w_d x_{nd})}{\partial w_k} \right)$$

Reordering

$$\frac{\partial E(w)}{\partial w_k} = - \sum_{n=1}^N x_{nk} \left(t_n - w_0 - \sum_{d=1}^D w_d x_{nd} \right)$$



2060756

Error Function Minimization

$$\frac{\partial E(w)}{\partial w_k} = - \sum_{n=1}^N x_{nk} \left(t_n - w_0 - \sum_{d=1}^D w_d x_{nd} \right)$$

$$\sum_{n=1}^N x_{nk} t_n = \sum_{n=1}^N x_{nk} \left(w_0 + \sum_{d=1}^D w_d x_{nd} \right) \rightarrow \underbrace{\sum_{n=1}^N x_{nk} t_n}_{\mathbb{X}^T t} = \underbrace{w_0 \sum_{n=1}^N x_{nk} + \sum_{d=1}^D w_d \sum_{n=1}^N x_{nk} x_{nd}}_{\mathbb{X}^T \mathbb{X} w}$$

Normal equations for the least squares problem

$$\mathbb{X}^T t = \mathbb{X}^T \mathbb{X} w$$

Multiplying both sides by $(\mathbb{X}^T \mathbb{X})^{-1}$

$$(\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T t = w$$

Takeaway $\rightarrow w^* = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T t$

Assuming $\mathbb{X}^T \mathbb{X}$ is invertible

Geometric Interpretation

- Linear Regression
- Basis Functions
- Vectorizing Calculations
- Error Function
- Error Function Minimization
- Geometric Interpretation
- Evaluation



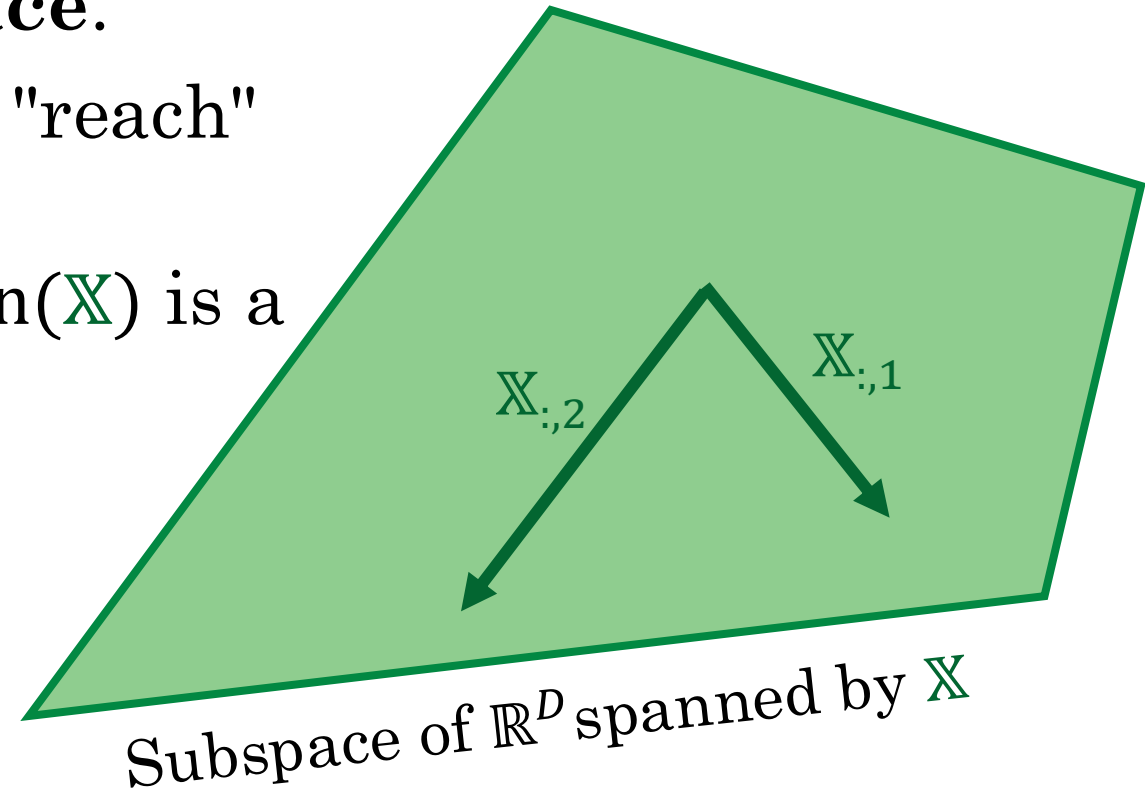
2060756



2060756

[Linear Algebra] Span

- The set of all possible linear combinations of the columns of $\mathbb{X}$ is called the **span** of $\mathbb{X}$ (denoted $\text{span}(\mathbb{X})$), also called the **column space**.
- Intuitively, this is all vectors you can "reach" using the columns of $\mathbb{X}$.
- If each column of $\mathbb{X}$ has length D , $\text{span}(\mathbb{X})$ is a subspace of $\mathbb{R}^D$.



[Linear Algebra] Matrix-Vector Multiplication



- **Approach 1:** So far, we've thought of our model as horizontally stacked predictions per datapoint:

$$\begin{bmatrix} | \\ \mathbb{Y} \\ | \end{bmatrix} = \begin{bmatrix} \text{---} & \mathbf{x}_1^T & \text{---} \\ \text{---} & \mathbf{x}_2^T & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{x}_N^T & \text{---} \end{bmatrix} \begin{bmatrix} w \end{bmatrix}$$

Matrix sizes
 $N \times 1 = (N \times (D + 1)) ((D + 1) \times 1)$

- **Approach 2:** However, it is helpful sometimes to think of matrix-vector multiplication as performed by columns. We can also think of $\mathbb{Y}$ as a **linear combination of feature vectors**, scaled by **parameters**.

$$\begin{bmatrix} | \\ \mathbb{Y} \\ | \end{bmatrix} = \begin{bmatrix} | & & | \\ \mathbb{X}_{:,0} & \dots & \mathbb{X}_{:,D} \\ | & & | \end{bmatrix} \begin{bmatrix} w \end{bmatrix} = w_0 \begin{bmatrix} | \\ \mathbb{X}_{:,0} \\ | \end{bmatrix} + w_1 \begin{bmatrix} | \\ \mathbb{X}_{:,1} \\ | \end{bmatrix} + \dots + w_D \begin{bmatrix} | \\ \mathbb{X}_{:,D} \\ | \end{bmatrix}$$



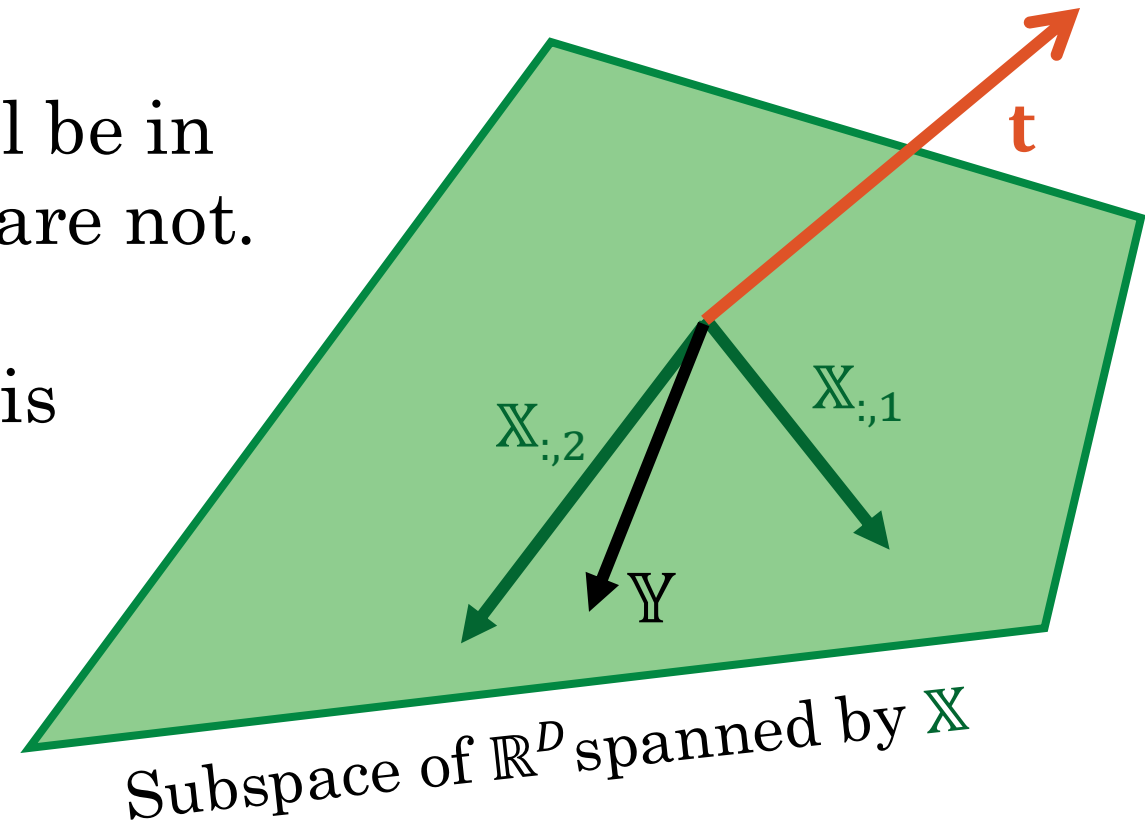
2060756

Prediction Is a Linear Combination of Columns

Our prediction of $\mathbb{Y}(\mathbb{X}, \vec{w}) = \mathbb{X} \vec{w}$ is a linear combination of columns of $\mathbb{X}$.

Interpret: Our linear prediction $\mathbb{Y}$ will be in $\text{span}(\mathbb{X})$, even if ground-truth values $\mathbf{t}$ are not.

Goal: Find vector of $\mathbb{Y}$ in $\text{span}(\mathbb{X})$ that is closest to $\mathbf{t}$.





**What's the geometry word for
'closest point in a subspace'?**

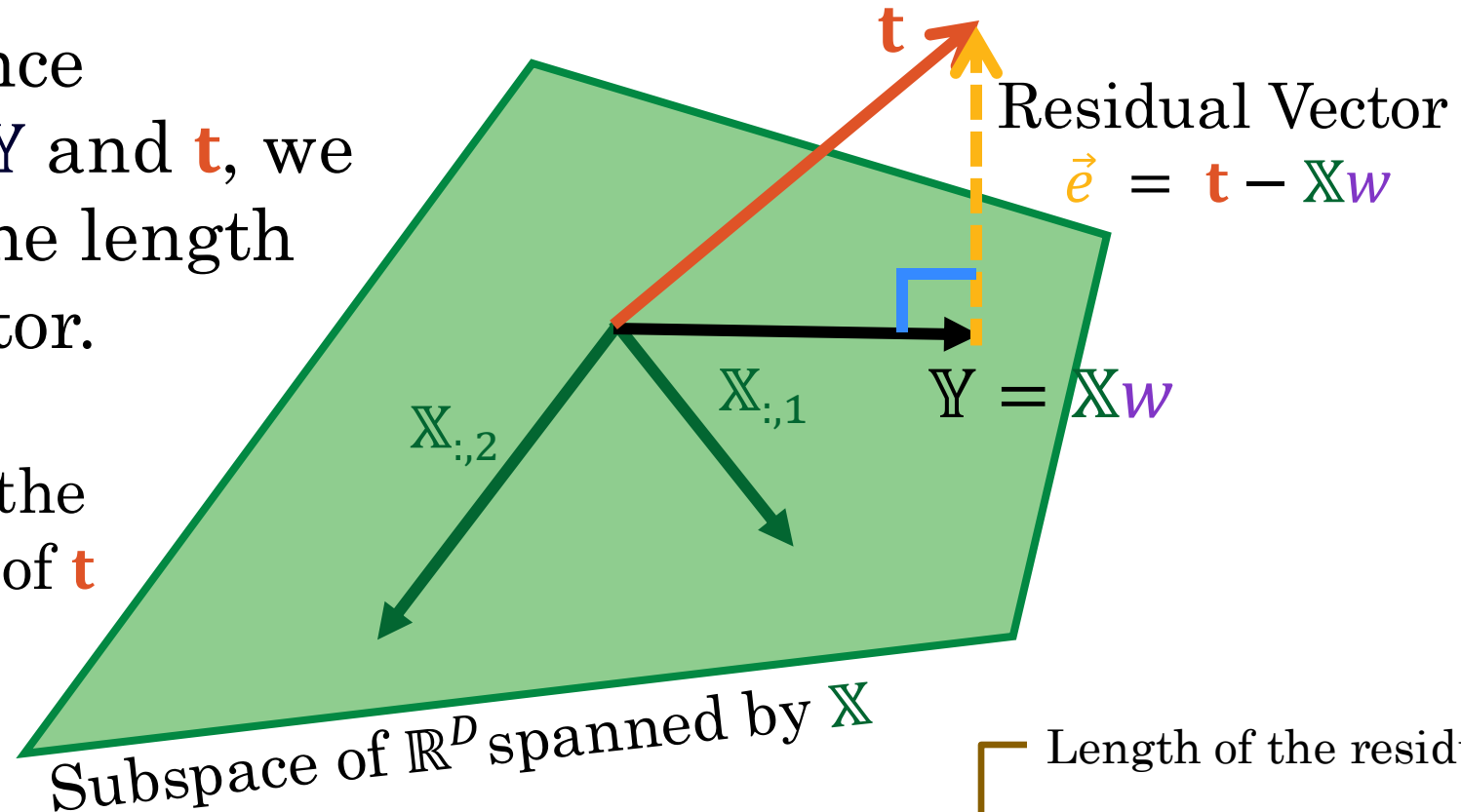


2060756

Finding Optimum Predictions ($\mathbb{Y}$)

To minimize distance between vector of $\mathbb{Y}$ and $\mathbf{t}$, we should minimize the length of the residual vector.

$\vec{e}$ is minimized if it is the orthogonal projection of $\mathbf{t}$ on the $\text{span}(\mathbb{X})$.



Length of the residual vector

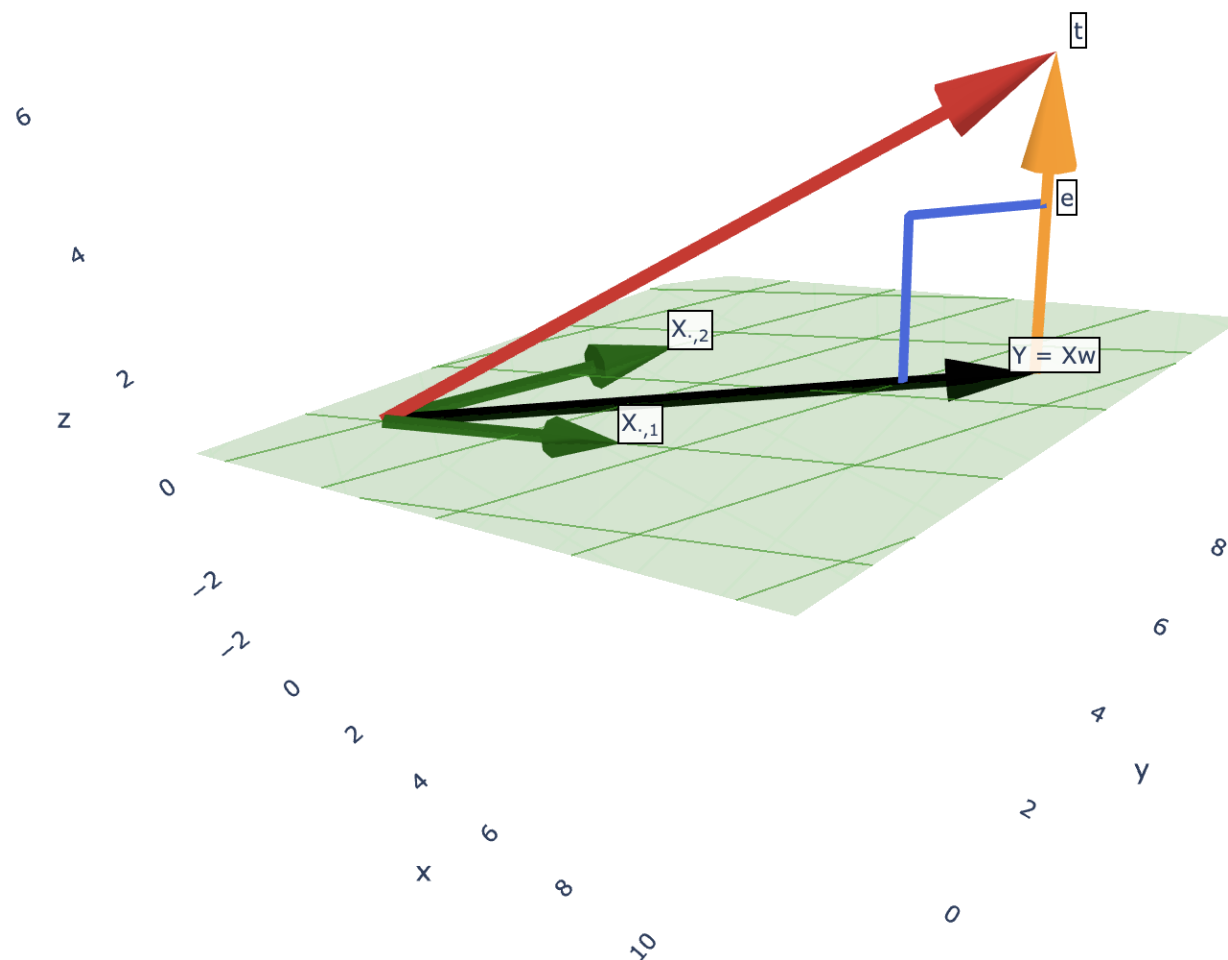
$$\begin{aligned} L_2 - \text{norm}(\vec{e}) &= \|\vec{e}\|_2 \\ &= \sqrt{\sum_{d=0}^D e_d^2} \end{aligned}$$

Geometry of Least Squares in Plotly

[Interactive link](#)



2060756

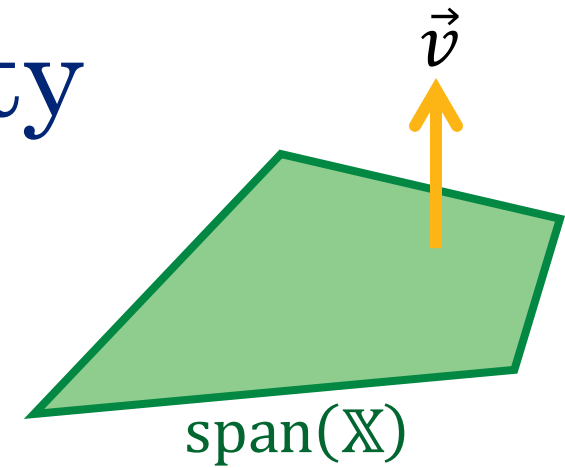


[Linear Algebra] Orthogonality



2060756

- A vector $\vec{v}$ is orthogonal to the $\text{span}(\mathbb{X})$ if and only if it is orthogonal to all columns of $\mathbb{X}$.



Assuming $\vec{v} \in \mathbb{R}^N$ and $\mathbb{X} = \begin{bmatrix} | & & | \\ \mathbb{X}_{:,0} & \dots & \mathbb{X}_{:,D} \\ | & & | \end{bmatrix} \in \mathbb{R}^{N \times D}$

$\vec{v}$ orthogonal to each column means:

$$\mathbb{X}_{:,0}^T \vec{v} = 0$$

$$\mathbb{X}_{:,1}^T \vec{v} = 0$$

$$\vdots$$

$$\mathbb{X}_{:,D}^T \vec{v} = 0$$

$$\begin{bmatrix} \mathbb{X}_{:,0}^T \vec{v} \\ \mathbb{X}_{:,1}^T \vec{v} \\ \vdots \\ \mathbb{X}_{:,D}^T \vec{v} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Rightarrow \mathbb{X}^T \vec{v} = 0$$



2060756

Going Back to Our Error Function

$$E(w) = \frac{1}{2} \sum_{n=1}^N (t_n - y(x_n, w))^2 = \frac{1}{2} (\|\vec{e}\|_2)^2$$

Equivalently, vector $\vec{e}$ is orthogonal to the $\text{span}(\mathbb{X})$, meaning that:

$$\mathbb{X}^T \vec{e} = 0 \quad \text{Adding the definition of residual}$$

$$\mathbb{X}^T (\mathbf{t} - \mathbb{Y}(\mathbb{X}, w^*)) = 0 \quad \text{Adding formula for } \mathbb{Y}(\mathbb{X}, w^*) = \mathbb{X}w^*$$

$$\mathbb{X}^T \mathbf{t} - \mathbb{X}^T \mathbb{X}w^* = 0 \quad \text{Moving terms}$$

$$\mathbb{X}^T \mathbf{t} = \mathbb{X}^T \mathbb{X}w^*$$

Normal
Equation

If $\mathbb{X}^T \mathbb{X}$ is
invertible

$$w^* = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbf{t}$$

Evaluation

- Linear Regression
- Basis Functions
- Vectorizing Calculations
- Error Function
- Error Function Minimization
- Geometric Interpretation
- Evaluation

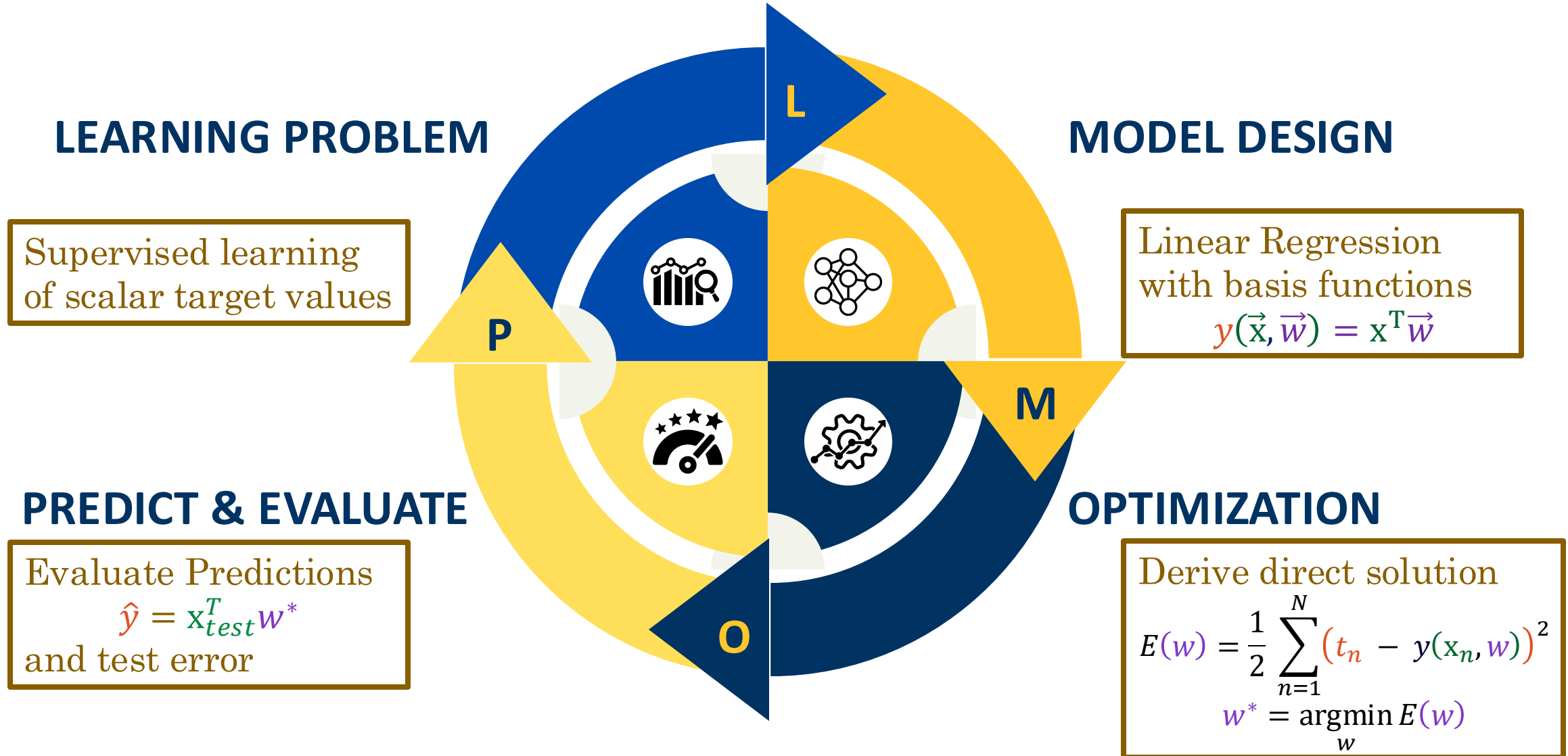


2060756

Predict and Evaluate



2060756





2060756

Evaluation - Visualization

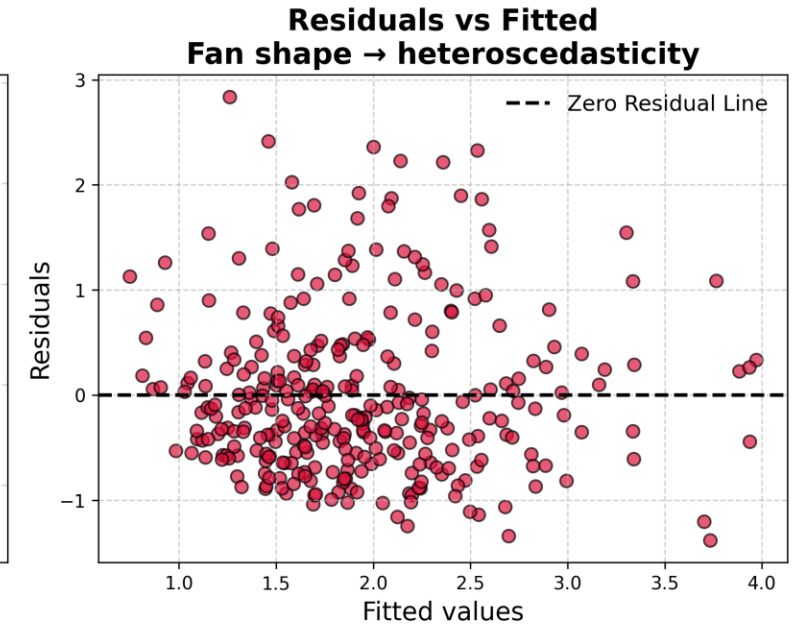
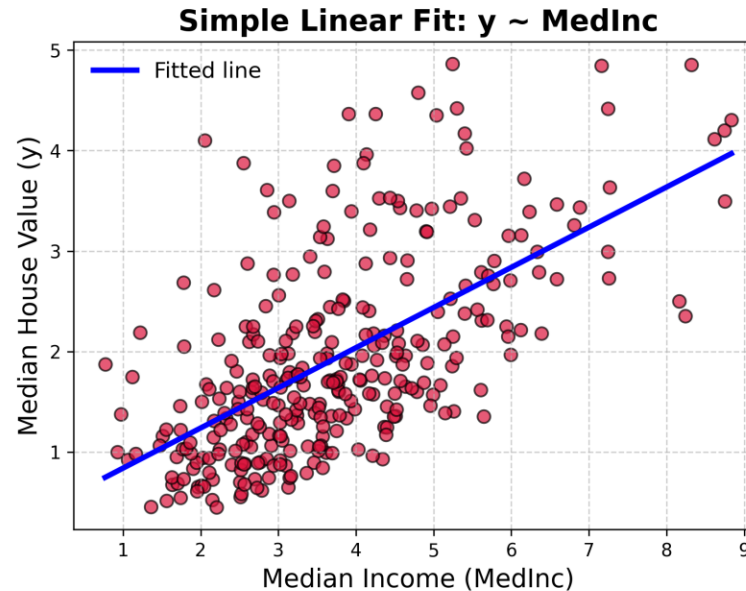
Residual plot shows the trend of the residuals $e_n = (t_n - y(x_n, w))$ with respect to predictions $y(x_n, w)$.

What a good residual plot looks like:

- Points scattered randomly around 0 (no pattern).

- Roughly constant vertical spread (homoscedasticity).

- No obvious trend with fitted values or with predictors.



Evaluation - Metrics



2060756

Evaluation - Metrics



2060756

Mean Squared Error (MSE)

$$\frac{1}{N} \left(\sum_{n=1}^N (t_n - y(\mathbf{x}_n, \mathbf{w}))^2 \right)$$

Evaluation - Metrics



Mean Squared Error (MSE)

$$\frac{1}{N} \left(\sum_{n=1}^N (t_n - y(\mathbf{x}_n, \mathbf{w}))^2 \right)$$

Root Mean Squared Error
(RMSE)

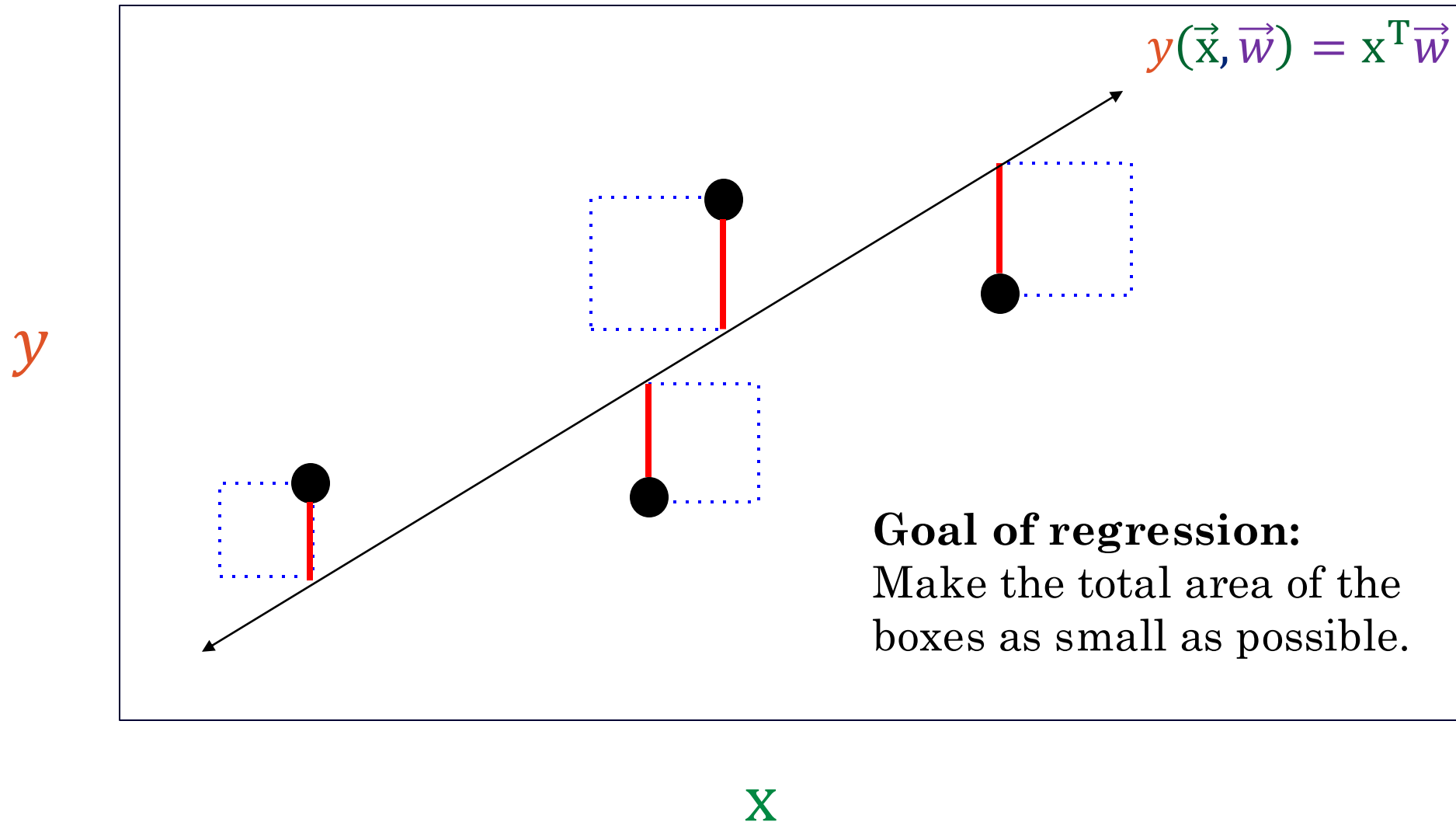
$$\sqrt{\frac{1}{N} \left(\sum_{n=1}^N (t_n - y(\mathbf{x}_n, \mathbf{w}))^2 \right)}$$

Moves the metric
back to the original
unit of the data
compared to MSE

Visualizing the Sum of Squared Error of Regression Model



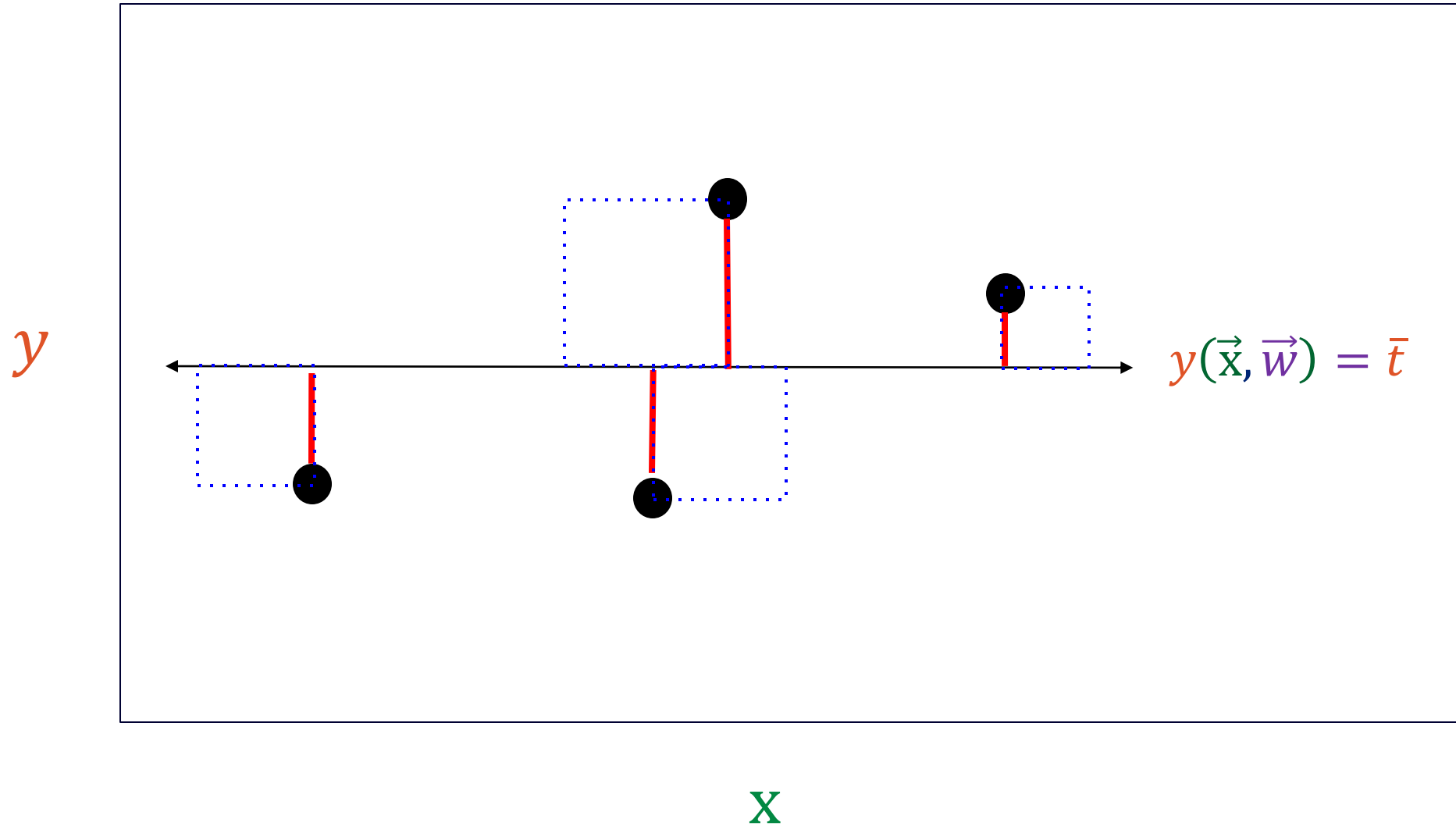
2060756



Visualizing the Sum of Squared Error of Intercept Model



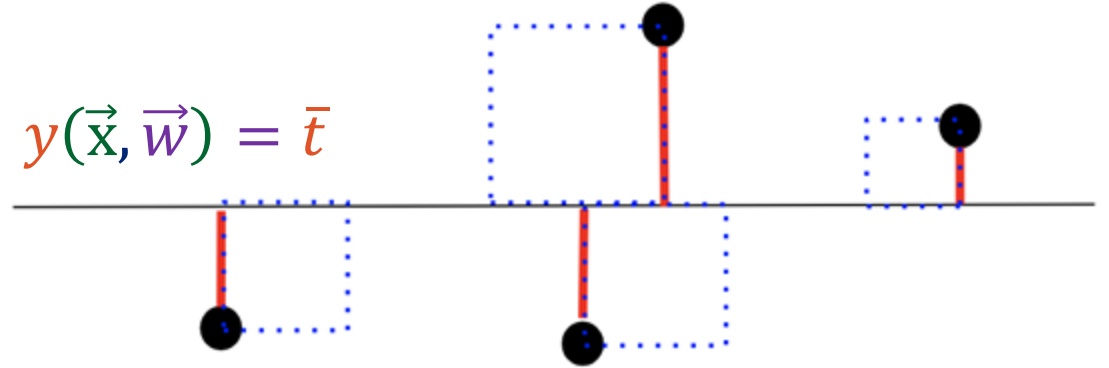
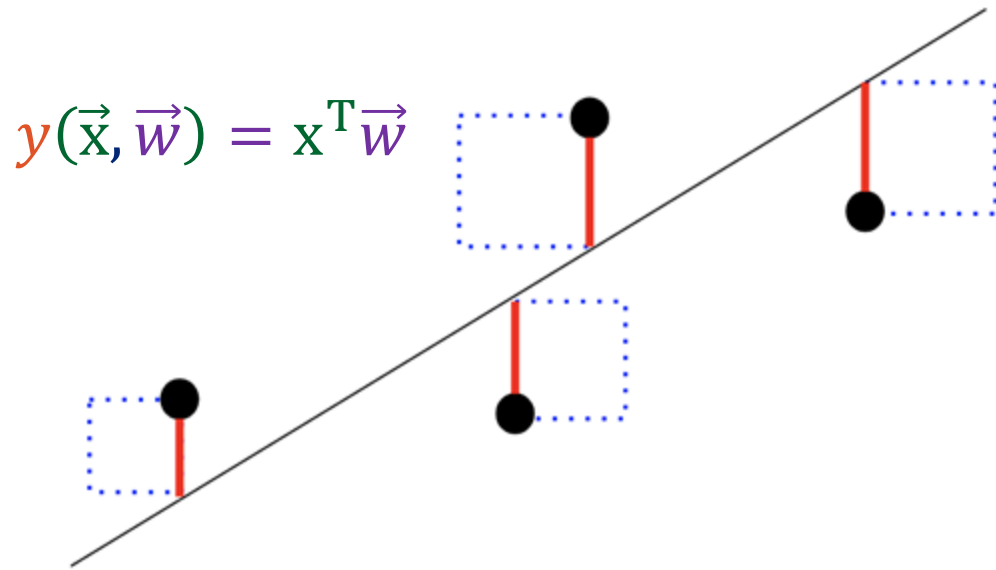
2060756



R²: Quality of the Fit Relative to Intercept Model



2060756



$$R^2 = \frac{\Delta \text{ in area}}{\text{Constant Model Area}} = 1 - \frac{\sum_{n=1}^N (t_n - y(x_n, w))^2}{\sum_{n=1}^N (t_n - \bar{t})^2}$$

unitless and only compares
performance relative to mean
baseline

Lecture 6

Linear Regression (1)

Credit: Joseph E. Gonzalez and Narges Norouzi

Reference Book Chapters: Chapter 1.2, Chapter 4.[1.4-1.6]