# Lecture 5. Linear regression

## COMP 551 Applied machine learning

Yue Li
Assistant Professor
School of Computer Science
McGill University

September 15, 2022

# Outline

Objectives

Simple linear regression

Hypothesis testing

Multiple linear regression

Non-linear basis function

Probabilistic interpretation

Summary

# Outline

Objectives

Simple linear regression

Hypothesis testing

Multiple linear regression

Non-linear basis function

Probabilistic interpretation

Summary

# Learning objectives

Understanding the following concepts

- Linear model
- Evaluation criteria
- Finding the best fit
- Geometric interpretation
- Maximum likelihood interpretation

# Outline

Objectives

Simple linear regression

Hypothesis testing

Multiple linear regression

Non-linear basis function

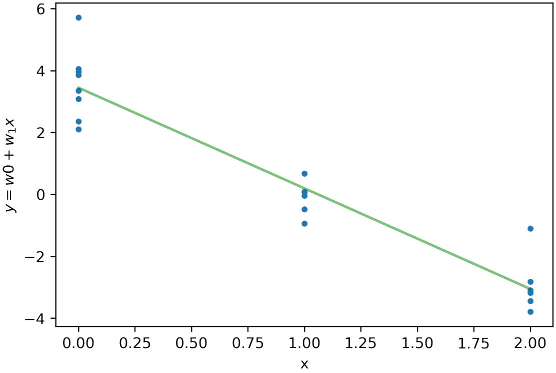
Probabilistic interpretation

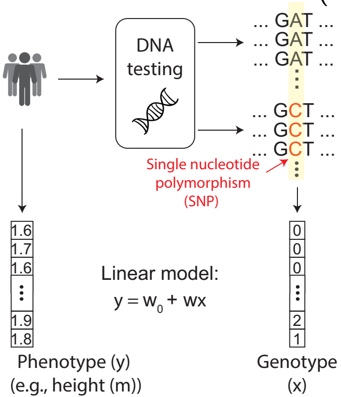
Summary

# Linear regression using a one-dimensional input

We want to predict real-valued quantity or often known as **response or target variable** $y \in \mathbb{R}$ by finding a mapping function that map from a **one-dimensional input x** to the real-valued $y$. We can fit a linear function on training examples $\{x_n, y_n\}_{n=1}^{N}$ (e.g., predicting phenotype from one genetic mutation):

$$f(x^{(n)}; w_0, w_1) = w_0 + w_1 x^{(n)} \tag{1}$$





# Simple linear regression using one input feature

$f(x^{(n)}; w_0, w_1) = w_0 + w_1 x^{(n)}$, where

- $w_0 = 3.4$ is the **intercept** or sometimes called bias in statistics, which is not be confused with the "model bias"
- $w_1 = -3.25$ is the **slope** of the linear function or the **regression coefficient**.

In statistics, we often write down the regression formula as:

$$y^{(n)} = w_0 + w_1 x^{(n)} + \epsilon^{(n)}$$

where $\epsilon^{(n)}$ is the prediction error for example $n$.

Using matrix notation, we can write the regression formula as:

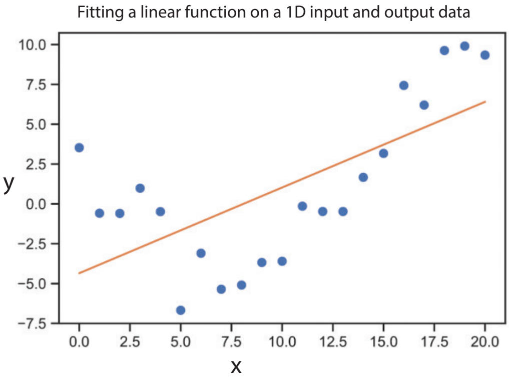
$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix} = \begin{bmatrix} x^{(1)} & 1 \\ x^{(2)} & 1 \\ \vdots & \\ x^{(N)} & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} + \begin{bmatrix} \epsilon^{(1)} \\ \epsilon^{(2)} \\ \vdots \\ \epsilon^{(N)} \end{bmatrix} \tag{2}$$

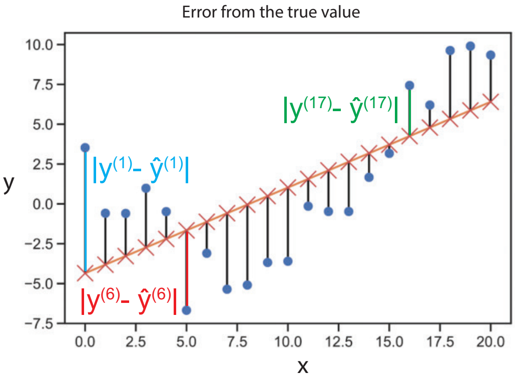
$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon} \tag{3}$$

# Residual error

Every straight line we fit incurs a prediction error on the training data point unless the line go through that data point. The **residual error** is Euclidean distance between the observed response $y^{(n)}$ value and the predicted value $\hat{y}^{(n)} = \mathbf{x}^{(n)}\mathbf{w}$:

$$l_n = ||y^{(n)} - \hat{y}^{(n)}||_2 = \sqrt{(y^{(n)} - \hat{y}^{(n)})^2} = |y^{(n)} - \hat{y}^{(n)}| \tag{4}$$





# Fitting a linear regression function by minimizing the sum of squared error

Sum of squared error (SSE) as a function of the linear coefficients **w** is defined as:

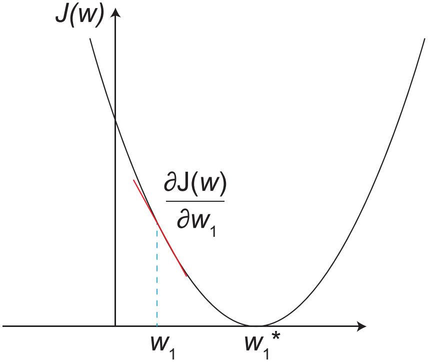
$$J(\mathbf{w}) = \sum_{n=1}^{N} (y^{(n)} - \hat{y}^{(n)})^2 = \frac{1}{2} \sum_{n=1}^{N} (y^{(n)} - w_0 - w_1 x^{(n)})^2 \tag{5}$$

**Goal**: find the best **w** to minimize SSE:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} J(\mathbf{w}) \tag{6}$$

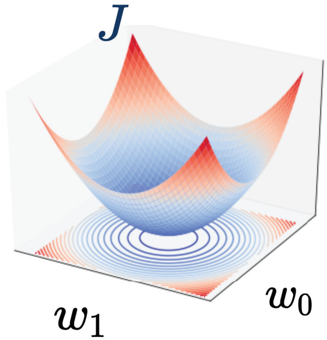
Given the error function is differentiable everywhere, the slope at the current value of $w_1$ projecting onto the error parabola as shown on the left.

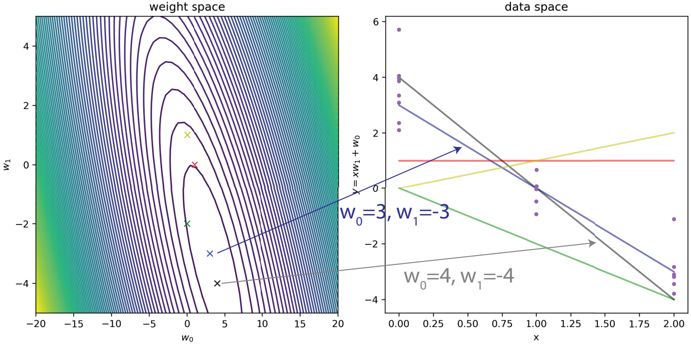
The SSE error function is **convex**. The optimal $w_1^*$ is found where **the slope or the partial derivative is zero** $\frac{\partial J(\mathbf{w})}{\partial w_1^*} = 0$.



# Visualization of the error surface as a function of both $w_1$ and $w_0$

$$J(\mathbf{w}) = \sum_{n=1}^{N}(y^{(n)} - \hat{y}^{(n)})^2 = \frac{1}{2}\sum_{n=1}^{N}(y^{(n)} - w_0 - w_1 x^{(n)})^2$$





$w_0$=3, $w_1$=-3

$w_0$=4, $w_1$=-4

# Derivation of ordinary least squared (OLS) solution for $w_1$ and $w_0$

$$\frac{\partial J(\mathbf{w})}{\partial w_1} = \frac{\partial}{\partial w_1}\sum_{n=1}^{N}(y^{(n)} - w_0 - w_1 x^{(n)})^2 = \sum_{n=1}^{N} 2(y^{(n)} - w_0 - w_1 x^{(n)})(-x^{(n)})$$

Set $\frac{\partial J(\mathbf{w})}{\partial w_1}$ to zero and multiplying $\frac{1}{2}$ and -1 on both sides gives:

$$\sum_{n=1}^{N}(y^{(n)} - w_0 - w_1 x^{(n)})x^{(n)} = 0 \tag{7}$$

We can do the same for the intercept $w_0$:

$$\frac{\partial J(\mathbf{w})}{\partial w_0} = \frac{\partial}{\partial w_0}\sum_{n=1}^{N}(y^{(n)} - w_0 - w_1 x^{(n)})^2 = \sum_{n=1}^{N} 2(y^{(n)} - w_0 - w_1 x^{(n)})(-1)$$

Set $\frac{\partial J(w_0)}{\partial w_0}$ to zero and multiplying $\frac{1}{2}$ and -1 on both sides gives:

$$\sum_{n=1}^{N}(y^{(n)} - w_0 - w_1 x^{(n)}) = 0 \tag{8}$$

# Derivation of ordinary least squared (OLS) solution for $w_1$ and $w_0$ (cont'd)

Solving for $w_0$:

$$
\begin{aligned}
\sum_{n=1}^{N} y^{(n)} - \sum_{n=1}^{N} w_0 - \sum_{n=1}^{N} w_1 x^{(n)} = 0 \\
\sum_{n=1}^{N} w_0 = \sum_{n=1}^{N} y^{(n)} - w_1 \sum_{n=1}^{N} x^{(n)} \\
N w_0 = \sum_{n=1}^{N} y^{(n)} - w_1 \sum_{n=1}^{N} x^{(n)} \\
w_0 = \frac{1}{N} \sum_{n=1}^{N} y^{(n)} - w_1 \frac{1}{N} \sum_{n=1}^{N} x^{(n)} \\
w_0 = \bar{y} - w_1 \bar{x}
\end{aligned}
$$

Plug the solution for $w_0$ into Eq. (8) $(\sum_{n=1}^{N}(y^{(n)} - w_0 - w_1 x^{(n)})x^{(n)} = 0)$ and solve for $w_1$:

$$\sum_{n=1}^{N}(y^{(n)} - (\bar{y} - w_1\bar{x}) - w_1 x^{(n)})x^{(n)} = 0$$

$$\sum_{n=1}^{N}(y^{(n)} - \bar{y} + w_1\bar{x} - w_1 x^{(n)})x^{(n)} = 0$$

$$\sum_{n=1}^{N}(y^{(n)} - \bar{y})x^{(n)} - w_1\sum_{n=1}^{N}(x^{(n)} - \bar{x})x^{(n)} = 0$$

$$w_1 = \frac{\sum_{n=1}^{N}(y^{(n)} - \bar{y})x^{(n)}}{\sum_{n=1}^{N}(x^{(n)} - \bar{x})x^{(n)}}$$

Note that

$$\sum_{n=1}^{N}(y^{(n)} - \bar{y})x^{(n)} = \sum_{n=1}^{N}(y^{(n)} - \bar{y})(x^{(n)} - \bar{x})$$

because:

$$\sum_{n=1}^{N}(y^{(n)} - \bar{y})(x^{(n)} - \bar{x})$$
$$= \sum_{n=1}^{N} y^{(n)}x^{(n)} - \sum_{n=1}^{N} y^{(n)}\bar{x} - \sum_{n=1}^{N}\bar{y}x^{(n)} + \sum_{n=1}^{N}\bar{y}\bar{x}$$
$$= \sum_{n=1}^{N} y^{(n)}x^{(n)} - N\bar{y}\bar{x} - \sum_{n=1}^{N}\bar{y}x^{(n)} + N\bar{y}\bar{x}$$
$$= \sum_{n=1}^{N}(y^{(n)} - \bar{y})x^{(n)}$$

Similarly,

$$\sum_{n=1}^{N}(x^{(n)} - \bar{x})x^{(n)} = \sum_{n=1}^{N}(x^{(n)} - \bar{x})(x^{(n)} - \bar{x})$$
$$= \sum_{n=1}^{N}(x^{(n)} - \bar{x})^2$$

# Update equations for the linear regression function

Therefore, the simple linear regression solutions are:

$$\hat{w}_1 = \frac{\sum_{n=1}^{N}(y^{(n)} - \bar{y})(x^{(n)} - \bar{x})}{\sum_{n=1}^{N}(x^{(n)} - \bar{x})^2}; \quad \hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x} \tag{9}$$

A special case arises when $\mathbf{y}$ and $\mathbf{x}$ are centered $\tilde{\mathbf{y}} = \mathbf{y} - \bar{y}$, $\tilde{\mathbf{x}} = \mathbf{x} - \bar{x}$ such that $\bar{y}' = 0$ and $\bar{x}' = 0$ (because $\sum \tilde{y}^{(n)} = \sum \mathbf{y} - \sum \bar{y} = N\bar{y} - N\bar{y} = 0$):

$$\hat{w}_1 = \frac{\sum_{n=1}^{N} \tilde{y}^{(n)} \tilde{x}^{(n)}}{\sum_{n=1}^{N} (\tilde{x}^{(n)})^2}; \quad \hat{w}_0 = 0$$

Without the intercept, the linear function becomes

$$f(\mathbf{x}; w_1) = w_1 \mathbf{x}$$

If we only center the input variable such that $\bar{x} = 0$ but not $\bar{y}$, then $w_0 = \bar{y}$ and the linear function $f(\mathbf{x}; w_1) = w_1 \mathbf{x} + \bar{y}$ measures the change of data point $n$ from the average $\bar{y}$ due to $w_1 x^{(n)}$.

# Standardization involves centering and scaling the input feature

Another special case arises if we further scale $\tilde{x}$ by its variance $s = \frac{1}{N}\sum_n (\tilde{x}^{(n)})^2$:

$$\mathbf{x} = \tilde{\mathbf{x}}/s$$

then we have

$$\sum_n (x^{(n)})^2 = \sum_n (\tilde{x}^{(n)}/s)^2 = \sum_n (\tilde{x}^{(n)})^2/s^2 = Ns^2/s^2 = N$$

As a result, the regression coefficients becomes further simplified as (while $w_0 = 0$ after centering input and response):

$$\hat{w}_1 = \frac{\sum_{n=1}^N \tilde{y}^{(n)}\tilde{x}^{(n)}}{\sum_{n=1}^N (\tilde{x}^{(n)})^2} = \frac{1}{N}\sum_{n=1}^N \tilde{y}^{(n)}\tilde{x}^{(n)} = \frac{1}{N}\mathbf{x}^\top \mathbf{y}$$

In the last equality, we take the inner product between the transposed $N \times 1$ vector $\mathbf{y}$ and the $N \times 1$ input vector $\mathbf{x}$.

Together, the procedure of centering response and/or input and scaling of the input are common practice known as after **standardization** mainly to simplify computation and making the model robust to different numerical scales of the input and response.

# Calculating the variance of $\mathrm{Var}[\hat{w}_1]$ and z-score

$$\mathrm{Var}[\hat{w}_1] = \mathrm{Var}[\frac{1}{N}\mathbf{x}^\intercal \mathbf{y}] = \frac{1}{N^2}\mathbf{x}^\intercal \mathrm{Var}[\mathbf{y}]\mathbf{x} \overset{\Delta}{=} \frac{1}{N^2}\mathbf{x}^\intercal \mathbf{x} \overset{\dagger}{=} \frac{1}{N}$$

- $^\Delta$The equality assumes that response is standardized and individuals are independent identically distributed (i.i.d.) such that $\mathrm{Var}[y] = I_{N \times N}$, which is an identity matrix of dimension $N \times N$.
- $^\dagger$The equality assumes that the input features is standardized such that $\mathbf{x}^\intercal \mathbf{x} = N$.

The **z-score** of the the regression coefficient is:

$$z_1 = \frac{\hat{w}_1}{\sqrt{\mathrm{Var}[\hat{w}_1]}} = \frac{\frac{1}{N}\mathbf{x}^\intercal \mathbf{y}}{\sqrt{\frac{1}{N}}} = \frac{\mathbf{x}^\intercal \mathbf{y}}{\sqrt{N}} \sim \mathcal{N}(0, 1)$$

# Outline

Objectives

Simple linear regression

Hypothesis testing

Multiple linear regression

Non-linear basis function

Probabilistic interpretation

Summary

# Hypothesis testing

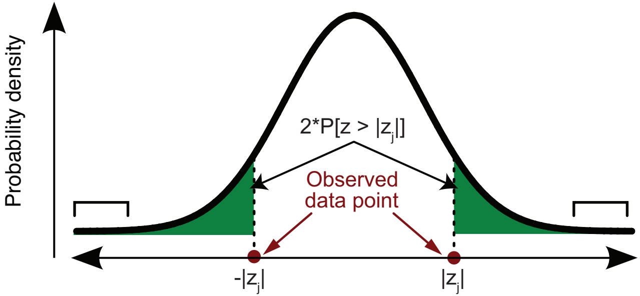
We test whether z-score is significantly different from the null distribution, which is a standard normal distribution $\mathcal{N}(0, 1)$:

Null hypothesis: $\mathcal{H}_0$: feature $x$ is not associated with the response $y$

Alternative hypothesis: $\mathcal{H}_1$: feature $x$ *is* associated with the response $y$

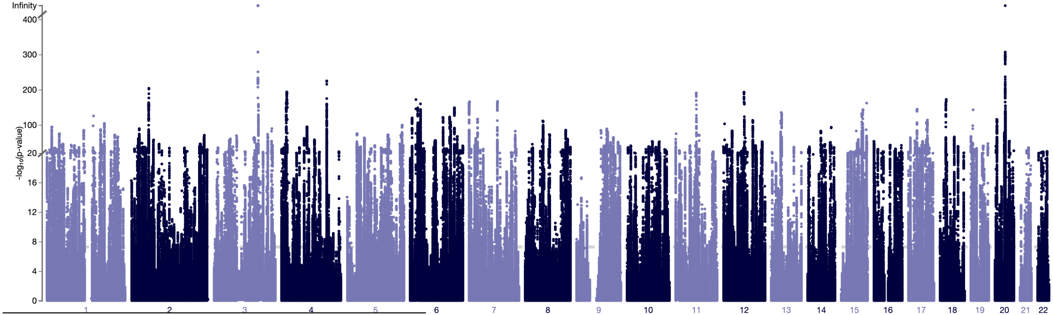
The two-sided p-value of the input feature association with response is then:

$$\text{p-value} = P[z < -|z_j|] + P[z > +|z_j|] = 2P[z > |z_j|]$$



# Displaying p-values for millions of mutations by Manhattan plot

- In our application of phenotype prediction using genotype, we can train one million linear regression models $\hat{w}_j$ each corresponding to $d^{th}$ mutation and test their significant association with a phenotype by comparing the z-score $z_d = \mathbf{x}^\top \mathbf{y}/N$ for $d \in \{1\ldots, D\}$ against the null $\mathcal{N}(0,1)$.
- Below plot is called *Manhattan plot*: x-axis is chromosomes 1-22; y-axis is $-\log_{10}$P-value computed for adult standing height of $N = 450,000$ individuals[1].
- To learn, consider taking COMP 565 this Winter.



[1] https://yanglab.westlake.edu.cn/data/ukb_fastgwa/imp/pheno/50

# Outline

Objectives

Simple linear regression

Hypothesis testing

Multiple linear regression

Non-linear basis function

Probabilistic interpretation

Summary

# Multiple linear regression

Motivations: predicting response using only one input feature is too limited. As we can see from the above standing height example, many phenotypes including diseases are associated with many genetic mutations.
Goal: learn how to fit a *multiple regression* to predict the outcome $y$ variable:

$$f(\mathbf{x}^{(n)};\mathbf{w}) = w_0 + w_1 x_1^{(n)} + w_2 x_2^{(n)} + \ldots + w_D x_D^{(n)} = w_0 + \sum_{d=1}^{D} w_d x_d^{(n)}$$

It is often convenient to add a feature $x_0$ equal to 1 across all training examples to treat the intercept as another regression coefficient:

$$f(\mathbf{x}^{(n)};\mathbf{w}) \equiv \hat{y}^{(n)} = w_0 x_0^{(n)} + w_1 x_1^{(n)} + w_2 x_2^{(n)} + \ldots + w_D x_D^{(n)} = \sum_{d=0}^{D} w_d x_d^{(n)}$$

# Multiple regression in matrix form

Suppose $N$ training examples and $D$ features. The data matrices are:

- Response: $\mathbf{y} \in \mathbb{R}^{N \times 1}$
- Input feature matrix: $\mathbf{X} \in \mathbb{R}^{N \times D}$
- Regression coefficients: $\mathbf{w} \in \mathbb{R}^{D \times 1}$

We can rewrite the multiple regression function as:

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \dots & x_D^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \dots & x_D^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & x_2^{(N)} & \dots & x_D^{(N)} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix} + \begin{bmatrix} \epsilon^{(1)} \\ \epsilon^{(2)} \\ \vdots \\ \epsilon^{(N)} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$$

Similar as in the simple regression, our goal is to find the OLS solution for the coefficients $\mathbf{w}$:

$$\mathbf{w}^* \leftarrow \arg\min_{\mathbf{w}} ||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})^\intercal(\mathbf{y} - \mathbf{X}\mathbf{w})$$

# Multiple regression OLS derivation

Let the loss function be $J(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^\intercal(\mathbf{y} - \mathbf{X}\mathbf{w})$. Instead of solving each coefficient one by one, we will solve them altogether using linear algebra.

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}(\mathbf{y} - \mathbf{X}\mathbf{w})^\intercal(\mathbf{y} - \mathbf{X}\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}}(\mathbf{y}^\intercal\mathbf{y}\underbrace{-\mathbf{y}^\intercal\mathbf{X}\mathbf{w} - \mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{y}}_{2\mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{y}}+\mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{X}\mathbf{w})$$

$$= \frac{\partial}{\partial \mathbf{w}}(\mathbf{y}^\intercal\mathbf{y} - 2\mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{y} + \mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{X}\mathbf{w}) = \underbrace{\frac{\partial}{\partial \mathbf{w}}\mathbf{y}^\intercal\mathbf{y}}_{0} -2\frac{\partial}{\partial \mathbf{w}}\mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{y} + \frac{\partial}{\partial \mathbf{w}}\mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{X}\mathbf{w}$$

$$= -2\mathbf{X}^\intercal\mathbf{y} + 2\mathbf{X}^\intercal\mathbf{X}\mathbf{w}$$

To get the last equality, we make use of two general properties in matrix differentiation:

$$\frac{\partial \mathbf{b}^\intercal\mathbf{A}}{\partial \mathbf{b}} = \mathbf{A}, \quad \frac{\partial \mathbf{b}^\intercal\mathbf{A}\mathbf{b}}{\partial \mathbf{b}} = \mathbf{2Ab}$$

Setting the derivative to zero and solve for $\mathbf{w}$ gives the closed-form solution:

$$0 = -2\mathbf{X}^\intercal\mathbf{y} + 2\mathbf{X}^\intercal\mathbf{X}\mathbf{w} \quad \rightarrow \quad \mathbf{X}^\intercal\mathbf{X}\mathbf{w} = \mathbf{X}^\intercal\mathbf{y} \quad \rightarrow \quad \mathbf{w} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y}$$

# Uniqueness of the OLS solution

$$\mathbf{w}^* = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y} \tag{10}$$

- The OLS solution is available when the $D \times D$ square matrix $\mathbf{A} = \mathbf{X}^\intercal\mathbf{X}$ is invertible.
- $\mathbf{A}$ is not invertible if some of its eigenvalues are zeros, which can happen when two features are perfectly correlated, e.g., $x_2 = 1 - x_1$
- At practice, $\mathbf{A}$ is not invertible often because either the sample size is smaller than the feature dimension $N << D$ or the samples are not i.i.d..

# Properties of the OLS solution (not critical for this course)

As the second last step in the above derivation, we obtain **normal equation**:

$$\mathbf{X}^\intercal \mathbf{X}\mathbf{w} = \mathbf{X}^\intercal \mathbf{y} \tag{11}$$

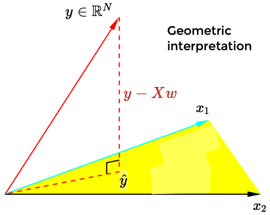
This is because the residual error $\mathbf{y} - \mathbf{X}\mathbf{w}$ is perpendicular to the space spanned by $\mathbf{X}$:

$$\mathbf{X}^\intercal \mathbf{X}\mathbf{w} = \mathbf{X}^\intercal \mathbf{y} \quad \rightarrow \quad \mathbf{X}^\intercal \mathbf{y} - \mathbf{X}^\intercal \mathbf{X}\mathbf{w} = 0 \quad \rightarrow \quad \mathbf{X}^\intercal(\mathbf{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$$

Plugging our OLS solution $\hat{\mathbf{w}} = (\mathbf{X}^\intercal \mathbf{X})^{-1}\mathbf{X}^\intercal \mathbf{y}$ back to linear classifier, we have

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}} = \mathbf{X}(\mathbf{X}^\intercal \mathbf{X})^{-1}\mathbf{X}^\intercal \mathbf{y} = \mathit{Proj}(\mathbf{X})\mathbf{y} \tag{12}$$

Here $\mathbf{X}(\mathbf{X}^\intercal \mathbf{X})^{-1}\mathbf{X}^\intercal$ is known as the "hat matrix" as it puts a hat on the response variable $\mathbf{y}$. Geometrically, it projects $\mathbf{y}$ onto the input space of $\mathbf{X}$.



# Time complexity

$$\mathbf{w}^* = (\mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}$$

- The inner product $A = \mathbf{X}^\intercal \mathbf{X}$ takes $O(D^2 N)$
- The inversion of the $D \times D$ matrix $A^{-1}$ takes $O(D^3)$
- Computing $\mathbf{X}^\intercal \mathbf{y}$ takes $O(ND)$

Therefore, the total time complexity is $O(ND^2 + D^3)$.

# Multivariate regression

We can also adapt the equation for multiple response variables. Instead of a response vector $\mathbf{y} \in \mathbb{R}^N$, we have a response matrix $\mathbf{Y} \in \mathbb{R}^{N \times K}$ for $K$ response variables.
The multivariate regression function is:

$$\mathbf{Y} = \mathbf{XW} \tag{13}$$

where $\mathbf{W} \in \mathbb{R}^{D \times K}$.
The OLS solution for $\mathbf{W}$ is then:

$$\mathbf{W} = (\mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{Y} \tag{14}$$

Note here the OLS coefficient $\mathbf{w}_k$ for each response variable $k$ is computed independently in the above solution. Therefore, the resulting OLS $\mathbf{W}$ is identical to fitting each $D \times 1$ regression coefficient $\mathbf{w}_k$ separately and then concatenate the K vectors together to form the matrix $\mathbf{W}$.

# Outline

Objectives

Simple linear regression

Hypothesis testing

Multiple linear regression

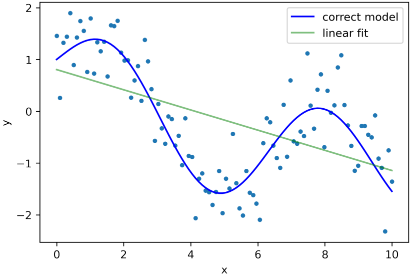
Non-linear basis function

Probabilistic interpretation

Summary

# Fitting non-linear data by transforming the input features

Consider the toy dataset below. It is obvious that our attempt to model $y$ as a linear function of $\hat{y} = w_0 + \mathbf{x}w_1$ would produce a bad fit.



**Idea**: we can create new more useful features using the given features. For example, we can create use a M-degree polynomial function and treat each power degree as a standalone feature:

$$\hat{y} = \mathbf{x}^0 w_0 + \mathbf{x}^1 w_1 + \mathbf{x}^2 w_2 + \ldots + \mathbf{x}^M w_M = \sum_{m=0}^{M} \mathbf{x}^m w_m = \phi(\mathbf{x})\mathbf{w}$$

# Fitting non-linear data by transforming the input features

More generally, we can transform the input $\mathbf{x}$ with a (non-linear) **basis function** $\phi(\mathbf{x})$. The multiple linear regression operates on the basis-transformed features:

$$\hat{y} = \sum_d w_d \phi_d(\mathbf{X}) = \Phi(\mathbf{X})\mathbf{w} \tag{15}$$

We then simply replace all of the occurrences of $\mathbf{X}$ with $\Phi(\mathbf{X})$ in the OLS solution:

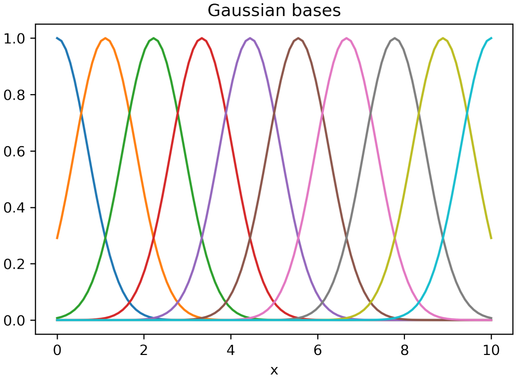
$$\hat{\mathbf{w}} = (\Phi(\mathbf{X})^\intercal \Phi(\mathbf{X}))^{-1} \Phi(\mathbf{X})^\intercal \mathbf{y}$$

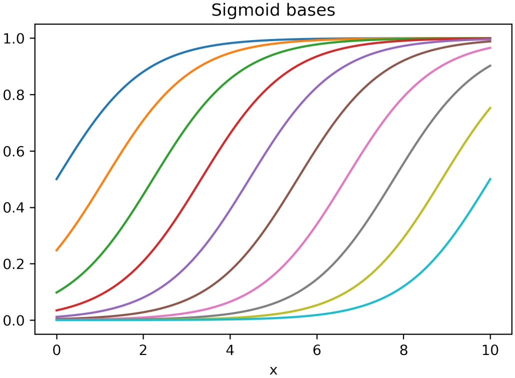
where

$$\Phi(\mathbf{X}) = \begin{bmatrix} \phi_1(\mathbf{x}^{(1)}) & \phi_2(\mathbf{x}^{(1)}) & \dots & \phi_D(\mathbf{x}^{(1)}) \\ \phi_1(\mathbf{x}^{(2)}) & \phi_2(\mathbf{x}^{(2)}) & \dots & \phi_D(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}^{(N)}) & \phi_2(\mathbf{x}^{(N)}) & \dots & \phi_D(\mathbf{x}^{(N)}) \end{bmatrix}$$

# Nonlinear basis functions

There are many nonlinear basis functions. Using scalar input $x \in \mathbb{R}$ as an example,



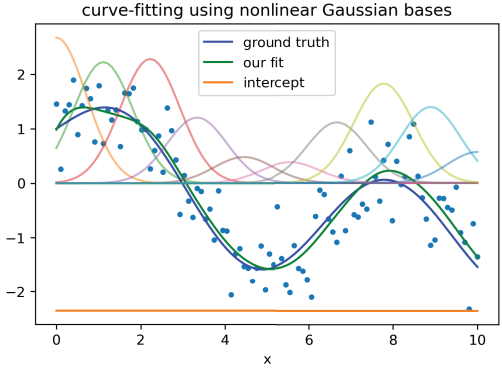


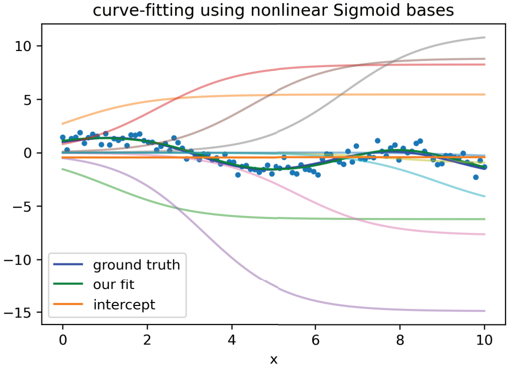
$$\phi_d(x) = \exp(-\frac{(x-\mu_d)^2}{s^2})$$

$$\phi_d(x) = \frac{1}{1+\exp(-\frac{(x-\mu_d)^2}{s})}$$

where each type of basis function has 10 different means $\mu_d \in [0, 10]$ and $s = 1$.

# Linear regression with nonlinear basis (See Colab for the implementations)





In both plots, the green curve (our fit) is the sum of these scaled Gaussian bases plus the intercept. Each of the 10 bases was scaled by the corresponding weight and displayed as individual colorful curves above.

$$\hat{y} = \sum_{d} w_d \phi_d(\mathbf{X})$$

# Outline

Objectives

Simple linear regression

Hypothesis testing

Multiple linear regression

Non-linear basis function

Probabilistic interpretation

Summary

# Probabilistic interpretation: Gaussian response variable

The general multivariate normal (MVN) distribution is:

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \Sigma) = \det(2\pi\Sigma)^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^{\intercal}\Sigma^{-1}(\mathbf{y} - \mathbf{X}\mathbf{w})\right) \tag{16}$$

where $\Sigma$ is a $N \times N$ covariance matrix between the $N$ samples. If we assume the samples are *i.i.d.*, then $\Sigma$ is a diagonal matrix $\sigma^2\mathbf{I}$, where $\mathbf{I}$ is an identity matrix:

$$\Sigma \overset{i.i.d.}{=} \begin{bmatrix} \sigma^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma^2 \end{bmatrix} = \sigma^2\mathbf{I}, \quad \Sigma^{-1} = \sigma^{-2}\mathbf{I}, \quad \det(2\pi\Sigma)^{-1/2} = (2\pi\sigma^2)^{-N/2}$$

The MVN can then be simplified as the product of individual Gaussians:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}) &= (2\pi\sigma^2)^{-N/2} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\mathbf{w})^{\mathsf{T}}(\mathbf{y}-\mathbf{X}\mathbf{w})\right) \\ &= \prod_n \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(y^{(n)} - \mathbf{x}^{(n)}\mathbf{w})^2\right) \end{aligned}$$

Taking the logarithm of the likelihood, we have

$$\begin{aligned} \ln p(\mathbf{y}|\mathbf{X}) &= \ln(2\pi\sigma^2)^{-N/2} + \left(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\mathbf{w})^{\mathsf{T}}(\mathbf{y}-\mathbf{X}\mathbf{w})\right) \\ &= \quad \underbrace{\frac{N}{\sigma\sqrt{2\pi}}}_{\text{constant w.r.t. } \mathbf{w}} \quad - \sum_n \left(\frac{1}{2\sigma^2}(y^{(n)} - \mathbf{x}^{(n)}\mathbf{w})^2\right) \\ &\propto -\frac{1}{2\sigma^2}\sum_n (y^{(n)} - \mathbf{x}^{(n)}\mathbf{w})^2 \end{aligned}$$

The last equation indicate that the log likelihood is proportional to the negative SSE.

# Maximum likelihood estimation

Given that,

$$\ln p(\mathbf{y}|\mathbf{X}) \propto -\frac{1}{2\sigma^2}\sum_n (y^{(n)} - \mathbf{x}^{(n)}\mathbf{w})^2 = -\frac{1}{2\sigma^2}J(\mathbf{w})$$

Since $\sigma^2$ is a constant, minimizing SSE w.r.t. **w** is equivalent to maximizing the Gaussian likelihood w.r.t. **w**:

$$\arg\min_{\mathbf{w}} J(\mathbf{w}) = \arg\max_{\mathbf{w}} \ln p(\mathbf{y}|\mathbf{X})$$

The maximum likelihood estimator for **w** is then identical to the OLS **w**:

$$\mathbf{w}^* = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y}$$

# Summary

- Response variable $y$ is modelled as a weighted linear combination function of the features $\hat{y} = \mathbf{X}\mathbf{w}$
- We fit the model by minimizing the sum of squared errors (SSE) $\sum_n (y^{(n)} - \mathbf{X}^{(n)}\mathbf{w})^2$
- OLS solution: $\mathbf{w}^* = (\mathbf{X}^\intercal \mathbf{X})^{-1}\mathbf{X}^\intercal \mathbf{y}$ with $O(ND^2 + D^3)$ time complexity
- z-score derived from the simple regression $\hat{y} = w_0 + \mathbf{x}w_1$ can be used to perform hypothesis testing for the statistical significance of the association between each feature (e.g., SNP) and the response (e.g., phenotype)
- Minimizing SSE is equivalent to maximizing the Gaussian likelihood given that the data points are $i.i.d.$