Revision

VECTORS, DOT PRODUCTS

Variance and Standard Deviation

- Both Indicate how spread out a data distribution is.

- Square root of variance is called Standard Deviation
  - A **low standard deviation** means that **data observations tend to be very close**
  - A **high standard deviation** indicates that **data observations are spread over a large range of values**

Correlation

- **Correlation** is a statistical term describing the degree to which two variables move in **coordination with one another**.
- If the two variables move in the same direction, then those variables are said to have a **positive correlation**.
- If they move in opposite directions, then they have a **negative correlation**.
- The strength of the correlation is determined by the **correlation coefficient**, which **varies between** $-\mathbf{1}$ **and** $+\mathbf{1}$.

What if correlation is 0?

Covariance

- Covariance measures the ***direction of the relationship between two variables***.

- A positive covariance means that ***both variables tend to be high or low at the same time.***

- A negative covariance means that ***when one variable is high, the other tends to be low***

Covariance vs Correlation Coefficient

- Covariance ***measures the direction of a relationship*** between two variables

- Correlation measures the ***strength of that relationship***.

- Both correlation and covariance are positive when the variables move in the same direction, and negative when they move in opposite directions.

- However, a correlation coefficient must always be $\boldsymbol{between\ -1\ and\ +1}$, with the extreme values indicating a strong relationship.

Fitting a Line (Recap)

- **In Statistics:**

$$\hat{y} = b_0 + b_1 X_1$$

In simple words, this is called "Regression coefficient". It tells how much impact of feature X_1 has on the output.

- **In Linear Algebra:**

$$\hat{y} = mx + b$$

What if we have more than one input/independent variables?

$$\hat{y} = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_k X_k$$

OR

$$\hat{y} = w_0 x_0 + w_1 x_1 + \cdots + w_n x_n$$

Remember this form of equation…

- $Slope = m = b_1$
- $y - intercept = b = b_0$

Before we start….

What we are going to revise?

- Lines
- Planes
- Hyperplanes
- Vectors

Why are we going to revise them?

- We are starting a series of algorithms which assume that the relationship between input features and output labels is **Linear** or **Kind of Linear.**
  - This holds true for both classification and regression.
  - We can always draw a straight line to separate classes, or to predict continuous output label.

Line

- A line can be represented in more than one ways…

$$\boldsymbol{y = mx + b}$$

Slope-intercept form

$$\boldsymbol{ax + by + c = 0 \text{ where } (a, b) \neq (0, 0)}$$

Standard Form of the line

Can we convert this standard form into vectors form?

We can suppose that (x_0, y_0) is a point on the line, then we can solve for c as:

$$\boldsymbol{ax_0 + by_0 + c = 0}$$

$$\boldsymbol{c = -ax_0 - by_0}$$

Now that we know what c equates to, we can put this in original equation as:

$$\boldsymbol{ax + by - ax_0 - by_0 = 0}$$

Rearrange:

$$\boldsymbol{ax - ax_0 + by - by_0 = 0}$$

$$\boldsymbol{a(x - x_0) + b(y - y_0) = 0}$$

This can e written s dot product of two vectors:

$$\boldsymbol{(a, b).(x - x_0, y - y_0) = 0}$$

Line

$$(\boldsymbol{a}, \boldsymbol{b}).(\boldsymbol{x} - \boldsymbol{x_0}, \boldsymbol{y} - \boldsymbol{y_0}) = \boldsymbol{0}$$

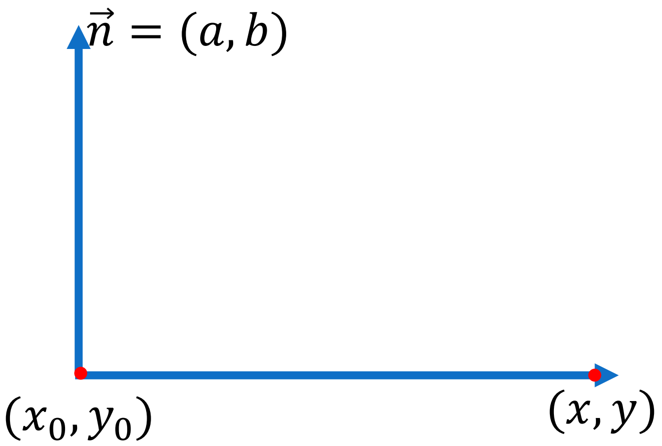
What should be the value of θ that satisfies this equation?

Recall how to find dot product between two vectors:

$$\vec{a}.\vec{b} = |a|.|b|cos\theta$$

If $\theta = 90°$ then $\vec{a}.\vec{b} = 0$ because cos of 90 is 0.

Therefore, to satisfy this equation, if the dot product of both vectors is 0 (given that $(a, b) \neq (0, 0)$), the two vectors must be perpendicular.



Line

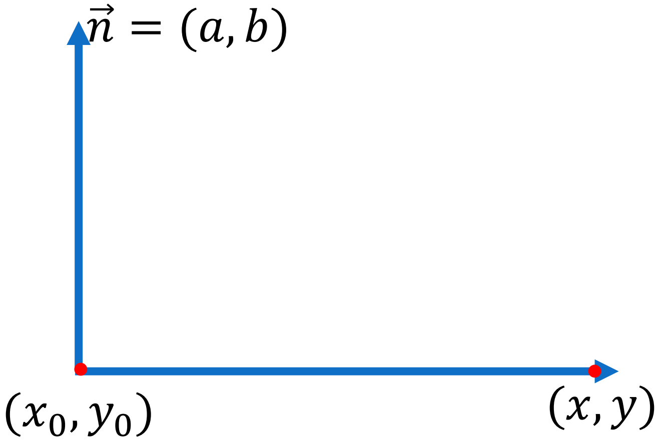
Let $\vec{n} = (a, b) \neq 0$ then:

$$\boxed{(a, b).(x - x_0, y - y_0) = 0}$$

Describes a line containing the point (x_0, y_0), perpendicular to the vector $\vec{n}$.

Thus, this is equivalent to the standard form of the line

$$\boxed{ax + by + c = 0 \text{ where } (a, b) \neq (0, 0)}$$



Line to Plane

Let $\vec{n} = (a, b) \neq 0$ then:

$$(a, b).(x - x_0, y - y_0) = 0$$

Describes a line containing the point (x_0, y_0), perpendicular to the vector $\vec{n}$.

We can extend the same notion to describe a plane, using a vector $\vec{n} = (\mathrm{a}, \mathrm{b}, \mathrm{c}) \neq 0$ and a point (x_0, y_0, z_0) on the plane:

$$(a, b, c).(x - x_0, y - y_0, z - z_0) = 0$$

Describes a plane containing the point (x_0, y_0, z_0), perpendicular to the vector $\vec{n}$.

This is equivalent to the standard form of the plane:

$ax + by + cz + d = 0$, **where** $(a, b, c) \neq (0, 0, 0)$

Can be further generalized for n dimensions.

Why are we interested in this form of line/plane when we have other much simpler forms?

Decomposition

- Recall this equation from Linear Regression:

$$\boldsymbol{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = 0}$$

- Can we achieve same result in the form of vectors?

$$\boldsymbol{(w_1, w_2, w_3, \dots, w_n).(x_1, x_2, x_3, \dots, x_n) = 0}$$

Which form is easier to implement in computers and why?

Can be thought of as a weight/importance/regression coefficient for its corresponding feature x_i

Food for thought: If somehow, we can **"control"** or **"learn"** these weights that minimize our error function as much as possible, it would be a great learning algorithm!

Hyperplane

- A Hyperplane is a subspace whose dimension is one less than that of its ambient space.
  - If a space is 3-dimensional, its hyperplanes are 2-dimensional **(Recall our discussion on decision boundaries)**
  - If the space is 2-dimensional, its hyperplanes are the 1-dimensional lines

$$\boxed{\boldsymbol{w_1x_1 + w_2x_2 + \cdots + w_nx_n = b}}$$

- When the coordinates are real numbers, this hyperplane **separates the space into two half-spaces** given by the inequalities:

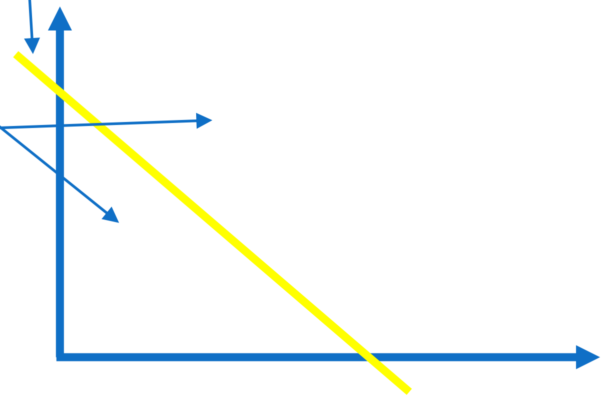
$$\boxed{\boldsymbol{w_1x_1 + w_2x_2 + \cdots + w_nx_n < b}}$$

$$\boxed{\boldsymbol{w_1x_1 + w_2x_2 + \cdots + w_nx_n > b}}$$

$$\boxed{\boldsymbol{(w_1, w_2, w_3, \dots, w_n).(x_1, x_2, x_3, \dots, x_n) < b}}$$

$$\boxed{\boldsymbol{(w_1, w_2, w_3, \dots, w_n).(x_1, x_2, x_3, \dots, x_n) > b}}$$

Some books use θ instead of w



Dot Product

- The dot product is commutative
  - For two vectors A and B

$$\boxed{\boldsymbol{A.B = B.A}}$$

$$\boxed{\boldsymbol{A.B = A^T B}}$$

$$\boxed{\boldsymbol{B.A = B^T A}}$$

Therefore:

$$\boxed{\boldsymbol{A^T B = B^T A}}$$

Making Maths Convenient

$$\boxed{w_1x_1 + w_2x_2 + \cdots + w_nx_n = b}$$

$$\boxed{w_1x_1 + w_2x_2 + \cdots + w_nx_n - b = 0}$$

For mathematical convenience, we often convert this form of linear expression to:

$$\boxed{w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n = 0}$$

Where $x_0 = 1$ and $w_0 = -b$ and the number of dimensions have been increased to $n + 1$

This allows us to just bundle the bias (intercept) into the expression and represent as:

$$\boxed{\sum_{i=0}^{n} w_i x_i = \vec{w}\,.\vec{x} = w^T x}$$

Where is $-b$?

It's absorbed in the equation as w_0 and x_0

This equation has a geometric interpretation as well….

Geometric Interpretation of Absorbing Bias (Intercept)

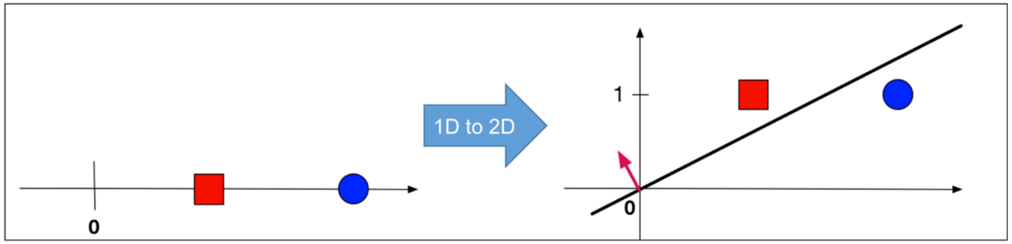
We can write this whole expression as:

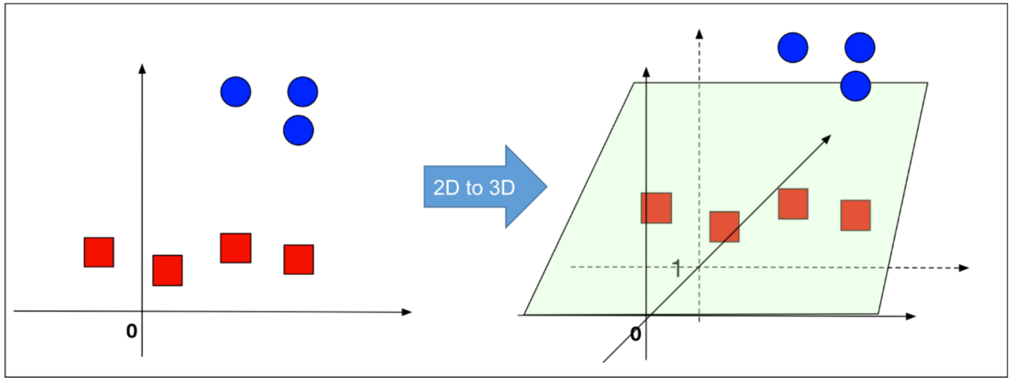
$$\boxed{\sum_{i=0}^{n} w_i x_i + 0 = 0}$$

Essentially, we are saying the intercept should be 0.

The plane that used to intercept at b, now must pass through origin!

We can claim that this new $n+1$ dimensional hyperplane always passes through the origin, but still, the old n-dimensional action takes place at $x_0 = 1$





Why is this useful?

When we know decision boundary always passes through origin, defining "weights vector" become easier!

Assignment 2: Task 1

- Use your images dataset to train a linear regression model.
  - Resize them just like before
- Use age as label.
- Compute R^2 on test split.
- Compute MSE or test split.

Book Reading

- Murphy – Chapter 1, Chapter 14
- Tom Mitchel (TM) – Chapter 3