# QBUS1040 - Notes :)

Foundations of Business Analytics (University of Sydney)

# QBUS1040



# VECTORS

## 1.1 VECTORS

Vectors → An ordered list of numbers

**Notation Styles:**

**OR**

**Names:**

- numbers in a vector → entries/coefficients
- length of vector → size/length/dimension
- numbers → scalars

*Examples*

- vector of length *n* is called an *n-vector*
- In the vector above has a dimension of 4 and a 3rd-entry value of 3.6

**Symbols:**

- Symbols that donate vectors:
- Other conventions:

- The i$^{th}$ element of a n-vector *a* is denoted as $a_i$
- For an n-vector, indices run from

*Examples*

If *a* is vector on previous slide,

Two vectors a and b are equal if

**Block Vectors**

If b, c and d are vectors with sizes m, n, p, the stacked vector or concentration of b, c, and d is:

*a* has a size of

**Zero, Ones and Unit Vectors**

- An n-vector with all entries 0 is denoted as
- An n-vector with all entries 1 is denoted by
- ***Unit Vector:*** has only one entry 1 and all others 0
- 1's location is denoted by

*Examples*

**Sparsity**

A vector is **sparse** is many of its entries are 0

 is the number of entries that are non-zero

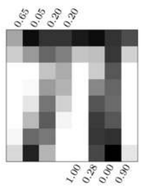
Examples are zero and unit vectors

**Practical Implications**

| | |
|---|---|
| **Location or displacement in 2-D/3-D** | 2 vector can represent a location or a displacement in 2-D   |

## Images & Video

**Monochrome (black and white) image**
grayscale values of $M \times N$ pixels stored as $MN$-vector (e.g., row-wise)




$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{62} \\ x_{63} \\ x_{64} \end{bmatrix} = \begin{bmatrix} 0.65 \\ 0.05 \\ 0.20 \\ \vdots \\ 0.28 \\ 0.00 \\ 0.90 \end{bmatrix}$$

**Color image**: $3MN$-vectors with R, G, B values of the $MN$ pixels

**Video**: $KMN$-vector represents $K$ monochrome images of $M \times N$ pixels

## Portfolio Vector

**Portfolio**

- $n$-vector represents stock portfolio or investment in $n$ assets
- $i$th element is amount invested in asset $i$
- elements can be number of shares, dollar values, or fractions of total dollar amount

$$p = (5000, 2000, 8000, 3000)$$

## Resource Vector

**Resource vector**

- elements of $n$-vector represent quantities of $n$ resources or commodities
- sign indicates whether quantity is held or owed, produced or consumed, ...
- example: bill of materials gives quantities needed to create a product

$$r = (\ -2,\quad 1,\quad 0,\quad -1)$$

## Feature Vectors

**Feature vectors**

contain values of variables or attributes that describe members of a set.

**Examples**

- age, weight, blood pressure, gender, ... of patients
- size, number of bedrooms, list price, ... of houses in an inventory

| Area ($m^2$) | Bedrooms | Bathrooms | Cars | Pets | Weekly rental price ($) |
|---|---|---|---|---|---|
| 252 | 3 | 2 | 2 | 1 | 1400 |

**Note**

- vector elements can represent very different quantities, in different units
- can contain categorical features (e.g., 0/1 for male/female)
- ordering has no particular meaning

## Word count Vectors

- a short document:

> **Word** count vectors are used **in** computer based **document** analysis. Each entry of **the word** count vector is **the number** of times **the** associated dictionary **word** appears **in the document**.

- a small dictionary (left) and word count vector (right)

$$\begin{array}{l} \text{word} \\ \text{in} \\ \text{number} \\ \text{horse} \\ \text{the} \\ \text{document} \end{array} \qquad \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \\ 4 \\ 2 \end{bmatrix}$$

- dictionaries used in practice are much larger

**Polynomials**

A polynomial with the degree of n-1:

can be represented by an n vector

*Example*

Polynomial form:

Vector Form:

## 1.2 VECTOR ADDITION

n-vectors a and b can be added, with sum denoted as a + b

*Example:*

**Rules of addition:**

- Commutative:
- Associative:
-
-

## 1.3 SCALAR – VECTOR MULTIPLICATION

Scalar can be multiplied:

*Example:*

**Rules of multiplication:**

- Associative:
- Left distributive:
- Right distributive:

**Linear Combinations**

For vectors and scalars   the *linear combination of vectors* is:

- The coefficients are:
- For any n-vector b:

## 1.4 INNER PRODUCT

The inner product of n-vectors a and b is:

*Example*

**Rules of inner product:**

- 
- 
- Commutative:
- Associative with scalar multiplication:
- Distributive with vector addition:
- 

**Simple formulas:**

- Inner product with *i*th unit vector picks out the *i*th entry:
- Differencing:
- Sum of entries:
- Average of entries:

*Example:*

- *w = vector of weights*
- *f = vector of features*
- 

## 1.5 COMPLEXITY OF VECTOR COMPUTATIONS

Computers store numbers in a ***floating point format***

Basic arithmetic operations (+, -, *) are called ***floating point operations / flops***

The complexity of an algorithm or operation = the total number of flops needed

Approximation of the time to execute:

*Average computers: 1Gflop/ second ($10^9$ flops / sec)*

**Complexity**

Operation count:

- Total number of operations in an algorithm
- (in linear algebra) a polynomial of the dimensions of the problem

Dominant Term → the highest order term in the flop count:

**Vector Operations:**

- for addition, subtraction: n flops
- for multiplication: n flops
- for inner product: flops
- (all of order n)

**Sub-vectors**

Sub Vectors are indicated by subscript:

# LINEAR FUNCTIONS, NORM AND DISTANCE

## LINEAR FUNCTIONS

- is a function mapping n-vectors to numbers
- If is an n-vector than is scalar
- NOTE: hence

*Example:*

Given the vector

**Superposition and Linear Functions**

- satisfies the superposition property if:

  For any numbers  and any vectors .

- A function that satisfies superposition is superposition is **linear.**
- IF f is linear than:

  For any n-vectors and any scalars

  *Example*

  Consider the function:

  *The function satisfies the superposition if applying the above function to  is the same as applying it to each of the terms.*

  If

  *Example*

  Does the function  satisfy the superposition property?

  *Example*

  Does the function  satisfy the superposition?

**The Inner Product Function**

with an n-vector, the function:

Is the inner product function

- is a weighted sum of the entries of x
- The inner product of the function is linear:

*all linear functions are inner products:

Suppose is linear: Then it can be expressed as for some a

Follows from:

*Examples in $R^3$*

is **linear:** with

is **linear:** with

is **not linear.**

**Affine function**

Affine Function → a function that is linear plus a constant is called affine.

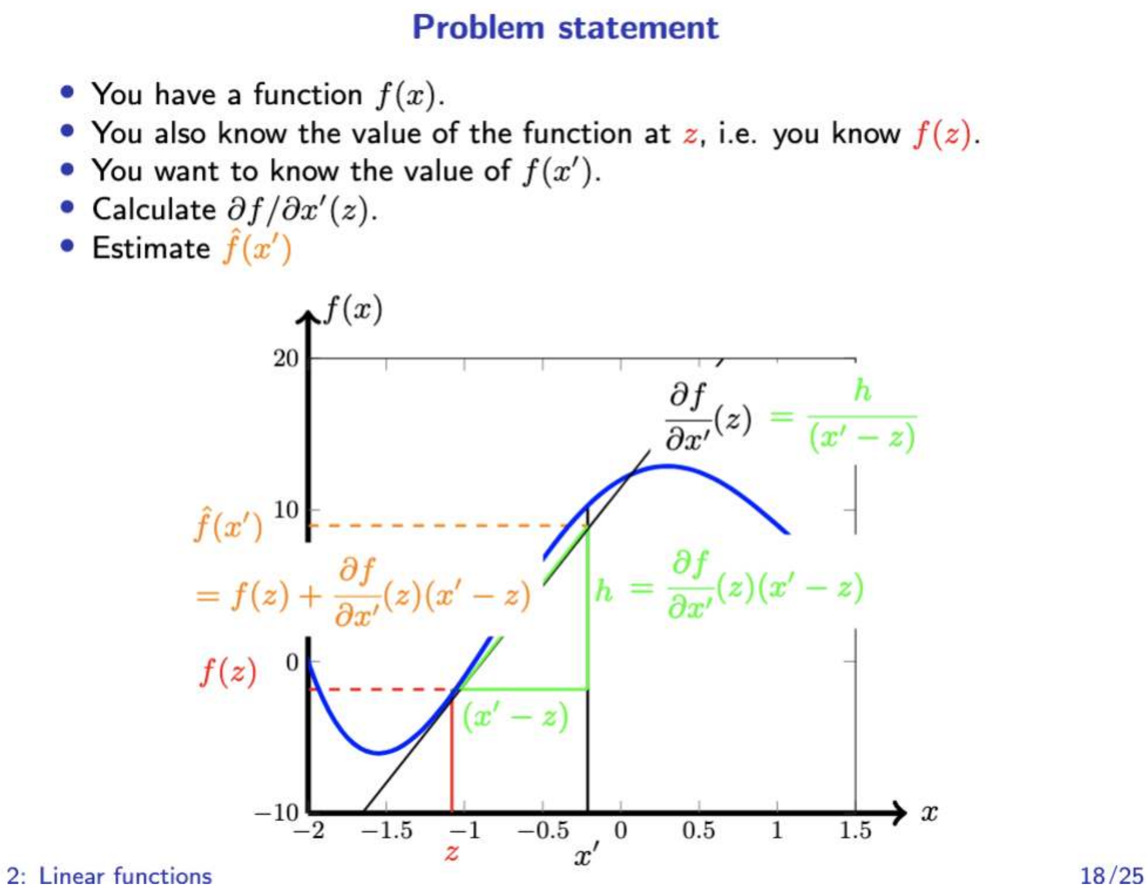
General form:

A function is affine ONLY if:

Holds for and all n-vectors

Every affine function can be written as

## AFFINE APPROXIMATION




Suppose

***First Order Taylor approximation*** *of ,* near point z:

is very close to  when  are all near

 is an affine function of x

You can write this using the inner product as:

n-vector =

= gradient of at

*Example with Two Variables*

*Calculate the Gradient:*

*First order Taylor approximation around*

## REGRESSION MODEL

Regression model = affine function of x

- = feature vector
- Elements in  are called regressors / independent variables / inputs
- = the vector of **weights** or **coefficients**
- Scalar = the offset / intercept
- Coefficients = **parameters** of the regression model
- = predication of some actual outcome or dependent variable, denoted

*Example:*

- = selling price of a house in $1000
- Regressor is:
- Regression model weight vector and offsets are:

## NORM AND DISTANCE

**Norm**

The Euclidian norm / norm of an n-vector  is:

- This is used to measure the **size** or **magnitude** of a vector
- If  reduces to its absolute bracket:
- For any n-vectors and any scalar :
  - Homogeneity:
  - Triangle inequality:
  - Nonnegativity:
  - Definiteness: only if:

**RMS Value**

The mean square of n-vector  is:

Root-mean-square value is:

- rms(x) gives the typical value of
- e.g., independent of n
- RMS value can be used to compare vectors of different dimensions

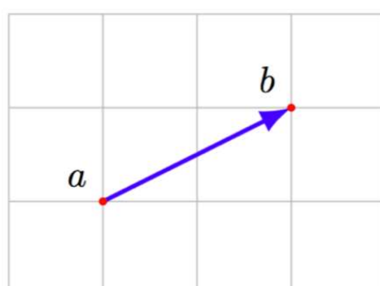
**Norm of block vectors**

- Suppose  are vectors:
- HENCE:

**Distance**

- Euclidean **distance** between n-vectors a and b is

agrees with ordinary distance for $n = 1, 2, 3$

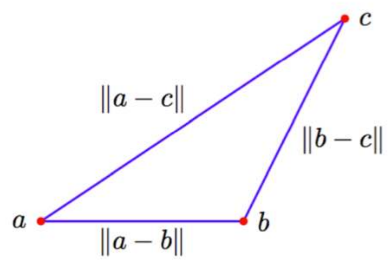



→ the RMS deviation between a and b

**Triangle Inequality:**

- Triangle with vertices at positions
- **Edges:**

**i.e.**, third edge length is no longer than the sum of other two

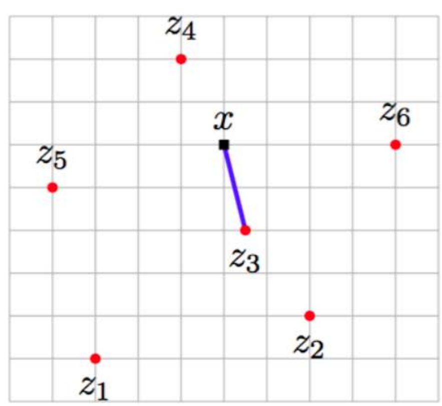



- 

**Feature Distance and Nearest Neighbours**

- If are feature vectors for two entities  is the **feature distance**
- If  = list of vectors, than  is the nearest neighbour of if:

$$\|x - z_j\| \leq \|x - z_i\|, \quad i = 1, \ldots, m$$




## 3. STANDARD DEV, ANGLES

### STANDARD DEVIATION

For n-vector x:

De- meaned vector is:

Standard deviation of x is:

gives the typical amount vary from

only if for some

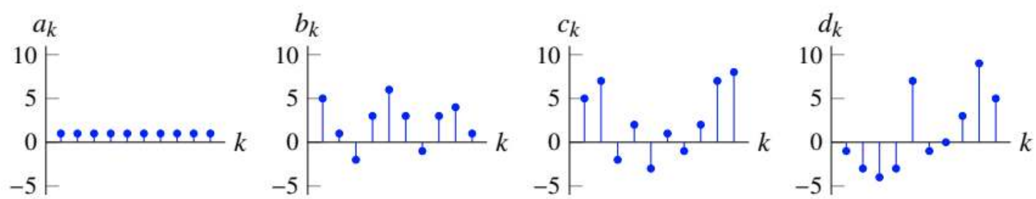
**Notation**

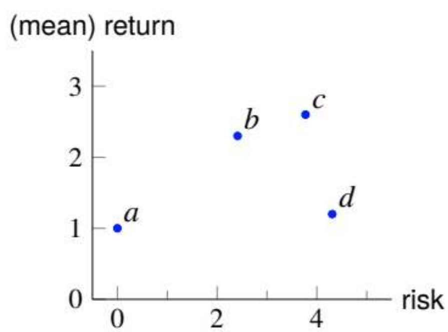
- Mean =
- Standard Deviation =

**Mean Return and risk**

- = time series of returns (e.g., with %) on some investment or asset over a period
- = the mean return over the period (return)
- = measures how variable the return is over the period (risk)

**Risk-return plot**




## ANGLE

**Cauchy-Schwarz inequality**

- For two n-vectors :
- Written in scalar form:

- Can also be written as
- To show the triangle inequality:

**Proof of Cauchy-Schwarz inequality**

- Assume , we show that

With equality only if: With equality only if:

For general nonzero apply the same argument to the unit norm vectors

The angle between two nonzero vectors defined as:

is the number inbetween that satisfies:

Coincides with ordinary angle between vectors in 2D and 3D

**Classification of angles**

If: are **orthorganal**, written

If: are **aligned**

If: are **anti-aligned**

If: make an **acute angle**

If: make an **obtuse angle**

**Correlation coefficient**

Consider vectors and their demeaned counterparts:

The correlation coefficient (between a and b with )

If: a and be are uncorrelated

I If: a and be are highly correlated

If: a and be are highly anti-correlated

is the average product of the deviations from the mean in standard units:

$$\rho = \frac{1}{n}\sum_{i=1}^{n}\frac{(a_i - \mathbf{avg}(a))}{\mathbf{std}(a)}\frac{(b_i - \mathbf{avg}(b))}{\mathbf{std}(b)}$$

- Highly correlated implies are typically both above (below) their means together

## CLUSTERING

Given

Goal: partition (divide, cluster) these vectors into groups (want vectors in the same group to be close to one another)

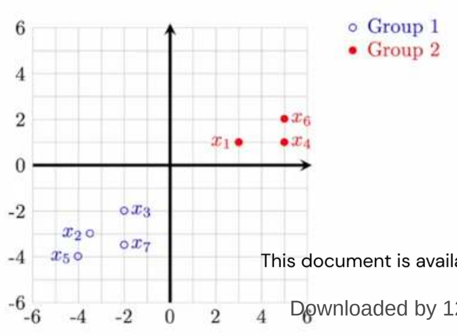
**Clustering Objective**

-
- is group
-
-

clustering objective is:

$$J^{\text{clust}} = \frac{1}{N}\sum_{i=1}^{N}\|x_i - z_{c_i}\|^2$$

**Example**



- N = Number of Data Points = 7
- *k* = Number of groups = 2
- G = elements in the group
-
- (order doesn't matter)
- C = which group the point is in
- E.g., because is in Group 1
- C can be displayed as a vector of the groups they are in:
-

- mean square distance from vectors to associated representative
- being small means good clustering
- **Goal:** choose clustering

## ALGORITHM

**Partitioning the vectors given representatives.**

- If representatives are given, how do we assign vectors to groups (aka, how do we choose ?)
- only appears in the term in

 Choose so that
 i.e. assign each vector to the nearest representative

**Choosing representatives given the particition**

- given the partition , how do we choose representatives to minimise ?
- splits into a sum of sums one for each :

$$J^{\text{clust}} = J_1 + \cdots + J_k, \quad J_j = \frac{1}{N} \sum_{i \in G_j} \|x_i - z_j\|^2$$

- We choose to minimise mean square distance to the points in its partition.
- This is the mean (or average or centroid) of the points in the partition.

$$z_j = \frac{1}{|G_j|} \sum_{i \in G_j} x_i$$

***k*-means algorithm**

- Alternate between updating the partition, then the representatives
- Objective: decreases in each step

**Given:** and

**Repeat:**

*Update partition: assign to*

*Update centroids:*

**Until** stops changing

**Convergence of k-means algorithm**

- goes down in each step until the stop changing.
- The k-means algorithm does not find the partition that minimises generally.
- K-means is a heuristic: it is not guaranteed to find the smallest possible value of
- The final partition and value of can depend on the initial representatives.

Common Approach

- Run the k-means 10 times, with different random initial representatives
- Take as a final partition the one with the smallest value of

## LINEAR INDEPENDENCE

A set of n-vectors with is linearly independent if:

Hold for some that are not all zero

- Equivalent to: at least is a linear combination of the others
- We say are linearly independent

Immediate observations:

- is linearly independent only if
- is linearly independent if and only if one is a multiple of the other.
- For more than two vectors, there is no simple way to state a condition

**Example**

The vectors:

Are linearly independent since

And you can express any of them as a linear combination of the other two, for example

A set of n vectors with is linearly independent if it is not linearly dependent

Holds only when

We say that  are linearly independent

No  is a linear combination of the others

**Linear combinations of linearly independent vectors**

Suppose x is linear combination of linearly independent vectors :

The coefficients are **unique –** if:

Then

This means we can deduce the coefficients from x

Hence:

## BASIS

**Independence Dimension inequality**

- A linearly independent set of n-vectors can have at most n elements
- Any set of n+1 of more n-vectors is linearly dependent
- A set of n linearly independent vectors is called a *basis*
- Any n-vector b can be expressed as a linear combination of them:
- These coefficients are unique
- The above formula is called: *an expansion of b in the  basis*

For example

 = basis

 = expansion of b

## ORTHONORMAL VECTORS

A set of n-vectors  are mutually orthogonal if  (at right angles/ perpendicular to each other)  for

They are normalised if

They are orthonormal if both hold

This can be expressed using inner products:

*If = 1, they are normalised*

Orthonormal sets of vectors are independent

By independence-dimension inequality, must have

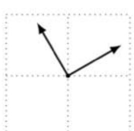
When  are an orthonormal basis

**Examples of orthonormal bases:**

- standard unit $n$-vectors $e_1, \ldots, e_n$
- the 3-vectors

$$\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \quad \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

- the 2-vectors shown below



**Orthonormal Expansion**

If  is an orthonormal basis, we have for any n-vector x:

This is called orthonormal expansion of x

To verify the formula, take inner product of both sides with

## GRAM-SCHMIDT ALGORITHM

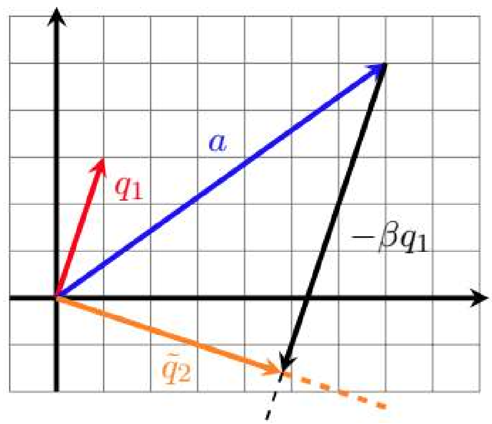
**Vector projection**

Consider the vectors

Goal: find a vector  where  and  such that we can construct

We can add the vectors

Hence:




Given the vectors

For

1. Orthogonalization:

2. Test for dependence

3. Normalization:

If G-S does NOT stop early in step 2,  are linearly independent

If G-S stops early in iteration  is a linear combination of  (is linearly dependent)

**Prove that  are orthonormal**

- for $i = 1$ we have $\tilde{q}_1 = a_1$ and $a_1 \neq 0$ (from independence of $a_i$s), so $q_1 = a_1/\|a_1\|$ is normalized
- for $i = 2$ assertion also holds: we have

$$\begin{aligned}\tilde{q}_2^T q_1 &= (a_2 - (q_1^T a_2)q_1)^T q_1 \\ &= a_2^T q_1 - (q_1^T a_2) q_1^T q_1 \\ &= 0\end{aligned}$$

- assume it's true for $i - 1$
- orthogonalization step ensures that

$$\tilde{q}_i \perp q_1, \ldots, \tilde{q}_i \perp q_{i-1}$$

- to see this, take inner product of both sides with $q_j$, $j < i$

$$\begin{aligned}q_j^T \tilde{q}_i &= q_j^T a_i - (q_1^T a_i)(q_j^T q_1) - \cdots - (q_{i-1}^T a_i)(q_j^T q_{i-1}) \\ &= q_j^T a_i - q_j^T a_i = 0\end{aligned}$$

- so $q_i \perp q_1, \ldots, q_i \perp q_{i-1}$
- normalization step ensures that $\|q_i\| = 1$

assuming G-S has not terminated before iteration $i$

- $a_i$ is a linear combination of $q_1, \ldots, q_i$:

$$a_i = \|\tilde{q}_i\| q_i + (q_1^T a_i) q_1 + \cdots + (q_{i-1}^T a_i) q_{i-1}$$

- $q_i$ is a linear combination of $a_1, \ldots, a_i$: by induction on $i$,

$$q_i = (1/\|\tilde{q}_i\|)(a_i - (q_1^T a_i) q_1 - \cdots - (q_{i-1}^T a_i) q_{i-1})$$

  and (by induction assumption) each $q_1, \ldots, q_{i-1}$ is a linear combination of $a_1, \ldots, a_{i-1}$

suppose G-S terminates in iteration $j$, i.e., $\tilde{q}_j = 0$

- from step 1 (orthogonalization):

$$0 = \tilde{q}_j = a_j - (q_1^T a_j) q_1 - \cdots - (q_{j-1}^T a_j) q_{j-1}$$

- hence $a_j$ is a linear combination of $q_1, \ldots, q_{j-1}$

$$a_j = (q_1^T a_j) q_1 + \cdots + (q_{j-1}^T a_j) q_{j-1}$$

- and each $q_1, \ldots, q_{j-1}$ is a linear combination of $a_1, \ldots, a_{j-1}$
- so $a_j$ is a linear combination of $a_1, \ldots, a_{j-1}$

**Complexity of the Gram Schmidt Algorithm**

Step 1 of iteration requires inner products

Which costs

Each

There are  subtractions.

Therefore are needed to compute

In step 3 we compute , which requires  to compute total is

$$\sum_{i=1}^{k}((4n-1)(i-1)+3n) = (4n-1)\frac{k(k-1)}{2} + 3nk \approx 2nk^2$$

using $\sum_{i=1}^{k}(i-1) = k(k-1)/2$