

# Midterm Key Points

## Similarity and Dissimilarity 相似性 & 相異性

### Hamming distance

- 相同位元長度的不同位元數量
- $Hamming\ distance = \sum_{i=1}^n 1(x_i \neq y_i)$
- Example
  - 001000
  - 110101
  - 0000x0
  - $Hamming\ distance = 5$

### Jaccard similarities

- 兩集合交集和聯集的比例
- $Jaccard\ similarities = \frac{|A \cap B|}{|A \cup B|}$
- Example
  - {1, 2, 3, 4}
  - {2, 3, 4, 5}
  - $Jaccard\ sim = \frac{|\{2,3,4\}|}{|\{1,2,3,4,5\}|} = \frac{3}{5} = 0.6$

### Jaccard distance

- $Jaccard\ distance = 1 - Jaccard\ similarities$
- Example
  - $Jaccarddis = 1 - 0.6 = 0.4$

### Cosine similarities

- 反映兩向量之間角度關係
- $Cosine\ similarities = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$

### Cosine distance

- $Cosine\ distance = 1 - Cosine\ similarities$

## Recommender Systems

## Content-based Recommendation ( 基於內容的推薦 )

- 優點：不需要大量使用者資料，推薦結果個性化。
- 缺點：過濾泡泡問題、依賴物品特徵、推薦多樣性較差。

## Collaborative Filtering ( 協同過濾 )

- 優點：無需物品特徵、推薦多樣性高。
- 缺點：冷啟動問題、資料稀疏問題。

## User-based vs Item-based Collaborative Filtering

- User-based：根據使用者相似性進行推薦。
- Item-based：根據物品相似性進行推薦。

## Clustering

### K-means vs. Hierarchical Agglomerative Clustering

- K-means 只需在每階段計算每一個點至重心的距離，階層式分群需在每階段都計算群與群的距離，在計算量而言，K-means 會有更好的效率

### K-means

- 重複分配以及更新直至收斂
  - 隨機選擇 **k** 個初始重心
  - 將每個點分配到最近的重心
  - 更新每群的重心
- **Euclidean Distance**
  - $distance(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$

### Hierarchical Agglomerative Clustering

- 每個點初始為單獨群集
- 將最近的兩群合併成一個新的群集
- 逐步合併最相近的群直到達到指定群數

## MapReduce

- Map Function: 將資料分成小單位進行處理
- Reduce Function: 對 Map Function 的輸出進行合併和計算
- Example
  - Map: 將每行文字分割為單詞，形成 (word,1) 的鍵值對

- Reduce: 計算每個單詞的總出現次數

## Dimensionality Reduction

### Singular Value Decomposition ( SVD )

- 將高維資料降維
- 用於資料降維和去除雜訊
- 保留重要特徵

## TF-IDF

- $TF = \frac{\text{出現次數}}{\text{文件總詞數}}$
- $IDF = \log \frac{\text{文章數}}{\text{包含詞的文章數}}$