Data Mining - Chapter 3: Similarity and Distances

- Based on "Data Mining: The Textbook" by Charu C. Aggarwal
- Presented by: [Tên Giảng viên]

Outline

- 3.1 Multidimensional Data
- 3.2 Text Similarity Measures
- 3.3 Temporal Similarity Measures
- 3.4 Graph Similarity Measures
- Exercises & Applications

3.1 Multidimensional Data

- Mỗi đối tượng: vector d chiều
- Dữ liệu số, chuẩn hóa có thể cần thiết
- Các độ đo chính:
  - **Khoảng cách Euclid** :
    - là một trong những phép đo khoảng cách phổ biến nhất, đại diện cho độ dài đường thẳng nối hai điểm trong không gian.
    - Khoảng cách Euclid càng nhỏ, hai đối tượng càng "gần" nhau hay càng tương đồng theo nghĩa hình học
  - Manhattan:

    là tổng giá trị tuyệt đối của sự khác biệt giữa các tọa độ. Tưởng tượng như bạn đi trong một thành phố với các khối nhà vuông vắn, bạn chỉ có thể đi dọc theo các con đường (ngang hoặc dọc), không thể đi chéo qua các tòa nh
  - Cosine similarity:
    - Đo lường góc giữa hai vector trong không gian đa chiều. Nó thường được sử dụng để xác định mức độ tương đồng về hướng của hai vector, không quan tâm đến độ lớn của chúng. Đây là phương pháp phổ biến trong xử lý ngôn ngữ tự nhiên (NLP) để so sánh các văn bản.

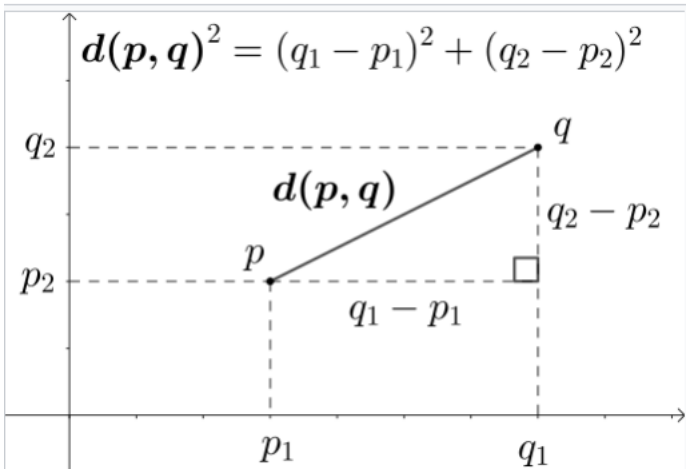
3.1 Multidimensional Data

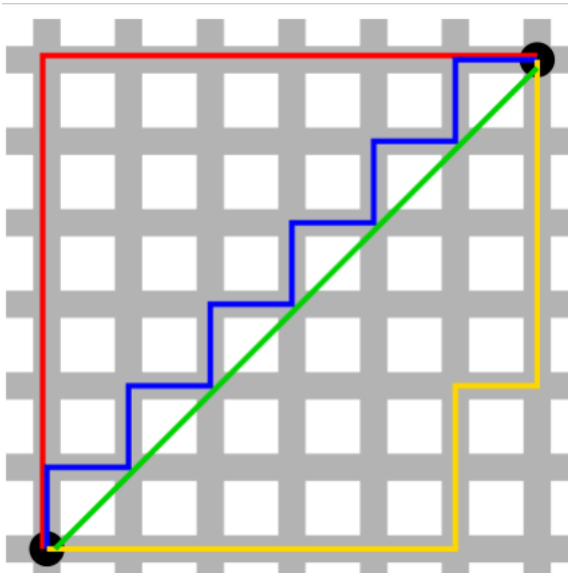
•Mỗi đối tượng được biểu diễn bằng một vector d chiều: x = (x_1, x_2, ..., x_d), y=
•Dữ liệu có thể cần chuẩn hóa để loại bỏ ảnh hưởng của đơn vị đo

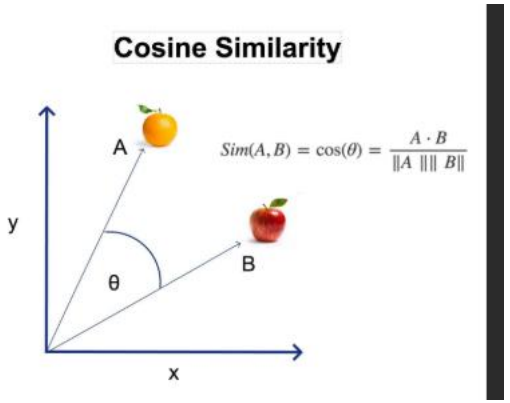
Các độ đo chính:

Các độ đo chính:

1. Euclidean distance: $d(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$
2. Manhattan distance: $d(x, y) = \sum_{i=1}^{d} |x_i - y_i|$
3. Cosine similarity: $\cos(\theta) = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}}$







Ví dụ tính toán

- A = (1, 2), B = (4, 6)
- Euclidean = $\sqrt{(1-4)^2+(2-6)^2} = \sqrt{9+16} = 5$
- Manhattan = $|1-4| + |2-6| = 7$
- Cosine = $\frac{(1*4+2*6)}{\sqrt{1^2+2^2}*\sqrt{4^2+6^2}} = \frac{16}{\sqrt{5}*\sqrt{52}}$

•**Khoảng cách Euclid / Manhattan:** Phù hợp khi các chiều dữ liệu có ý nghĩa về
•"khoảng cách" thực sự và bạn quan tâm đến sự khác biệt về độ lớn tuyệt đối giữa các giá trị. Khoảng cách Euclid nhạy cảm hơn với các chiều có sự khác biệt lớn.
•**Cosine Similarity:** Lý tưởng khi bạn quan tâm đến "hướng" hay "mô hình" của dữ liệu hơn là độ lớn tuyệt đối. Rất hữu ích khi các vector có độ lớn khác nhau nhưng lại có cùng cấu trúc hoặc phân bố (ví dụ: các tài liệu có độ dài khác nhau nhưng nói về cùng một chủ đề).

3.2 Text Similarity Measures

- Đo lường độ tương đồng văn bản là việc định lượng mức độ giống nhau về ngữ nghĩa hoặc cấu trúc giữa hai đoạn văn bản, câu, hay tài liệu
- Đặc điểm: văn bản là dữ liệu phi cấu trúc
- Biểu diễn:
  - Bag-of-Words (BoW)
  - TF-IDF
- Độ đo:
  - - Cosine similarity
  - - Jaccard similarity
  - - Edit/Levenshtein distance

Text Similarity Measures

- Văn bản là dữ liệu phi cấu trúc → chuyển thành vector (BoW, TF-IDF)

Các độ đo chính:

1. **Cosine similarity**: phổ biến nhất trong biểu diễn vector văn bản
2. **Jaccard similarity**: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$
3. **Edit (Levenshtein) distance**: số thao tác cần để biến đổi chuỗi A → B (thêm, xóa, thay ký tự)

Ví dụ Cosine similarity văn bản

- A: "data mining is useful"
- B: "mining data is very useful"

=>Vector BoW:

Từ vựng: [data, mining, is, useful, very]

A = [1, 1, 1, 1, 0], B = [1, 1, 1, 1, 1]

→ Vector BoW → Cosine(A, B)=

4/(sqtr(4)+sqrt(5)) ≈ 0.89

VD **Jaccard Similarity**

Văn bản 1: "Hôm nay trời đẹp, tôi đi chơi công viên.“

Văn bản 2: "Trời đẹp hôm nay, tôi thích đi công viên."

Bước 1: Chuẩn hóa và tạo tập hợp từ (loại bỏ từ dừng, chuyển về chữ thường, v.v. - cho đơn giản, bỏ qua bước này trong ví dụ):

Tập A (từ Vb1): {"Hôm", "nay", "trời", "đẹp", "tôi", "đi", "chơi", "công", "viên"}

Tập B (từ Vb2): {"Trời", "đẹp", "hôm", "nay", "tôi", "thích", "đi", "công", "viên"}

Bước 2: Tìm phần giao và hợp:

A∩B: {"Hôm", "nay", "trời", "đẹp", "tôi", "đi", "công", "viên"}

A∪B: {"Hôm", "nay", "trời", "đẹp", "tôi", "đi", "chơi", "công", "viên", "thích"}

Bước 3: Tính Jaccard Similarity: |A∩B|=8 |A∪B|=10

Jaccard(A,B)=10/8=0.8

Ý nghĩa: Giá trị 0.8 cho thấy hai văn bản có mức độ trùng lặp từ vựng khá cao, ám chỉ chúng có nội dung tương tự.

TF-IDF and Cosine Similarity (TF-IDF và Độ Tương Đồng Cosine)

- Đây là một trong những phương pháp phổ biến và mạnh mẽ nhất để đo độ tương đồng văn bản.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** Là một kỹ thuật thống kê dùng để đánh giá tầm quan trọng của một từ trong một tài liệu so với một tập hợp tài liệu.
- **TF (Term Frequency):** Tần suất xuất hiện của một từ trong một tài liệu cụ thể.

$$\text{TF}(t, d) = \frac{\text{Số lần từ } t \text{ xuất hiện trong tài liệu } d}{\text{Tổng số từ trong tài liệu } d}$$

- **IDF (Inverse Document Frequency):** Đo lường mức độ phổ biến hay hiếm của một từ trong toàn bộ tập hợp tài liệu. Từ càng hiếm thì IDF càng cao.

$$\text{IDF}(t, D) = \log\left(\frac{N}{\text{df}(t)}\right)$$

Trong đó:

- N: Tổng số tài liệu trong tập hợp D.
- $\text{df}(t)$: Số tài liêu chứa từ t.

- **TF-IDF score:**

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

VD TF-IDF

- **Ví dụ Minh Họa:**
  - D1: "tôi thích học máy"
  - D2: "tôi học AI"
  - D3: "học máy là AI"
  - **Bước 1: Tính TF-IDF cho từng từ trong D1 và D2 (ví dụ).**
    - TF-IDF("tôi", D1) $\approx$ 0.059, TF-IDF("thích", D1) $\approx$ 0.159, TF-IDF("học", D1) = 0, TF-IDF("máy", D1) $\approx$ 0.059
    - TF-IDF("tôi", D2) $\approx$ 0.059, TF-IDF("học", D2) = 0, TF-IDF("AI", D2) $\approx$ 0.088
  - **Bước 2: Biểu diễn D1 và D2 dưới dạng vector TF-IDF.**
    - Vector D1 (theo thứ tự từ điển): (0.059, 0.159, 0, 0.059, 0, 0)
    - Vector D2 (theo thứ tự từ điển): (0.059, 0, 0, 0, 0.088, 0)
  - **Bước 3: Tính Cosine Similarity giữa D1 và D2.**
    - $\text{similarity}(D1, D2) \approx 0.183$ (tính toán chi tiết ở phần trước).
- **Ý nghĩa:** Giá trị thấp (0.183) cho thấy D1 và D2 không thực sự tương đồng về chủ đề chính, dù có một số từ chung.

3.3 Temporal Similarity Measures

- Áp dụng: chuỗi thời gian (thời tiết, cổ phiếu,...), So sánh các chuỗi dữ liệu (time series) hoặc sự kiện diễn ra theo thời gian, có tính đến cả giá trị và thứ tự thời gian.
- - Euclidean distance (cứng)
- - Dynamic Time Warping (DTW)
- - Pearson Correlation (quan hệ tuyến tính)

1. **Euclidean**:
   - Nhạy với độ lệch thời gian nhỏ
2. **Dynamic Time Warping (DTW)**:
   - So khớp hai chuỗi bằng cách co giãn thời gian
   - Độ phức tạp: $O(n^2)$
3. **Pearson Correlation Coefficient**: $r = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2}\cdot\sqrt{\sum(y_i-\bar{y})^2}}$

Dynamic Time Warping (DTW)

Giả sử hai chuỗi thời gian:

- $TS_1 = (x_1, x_2, \ldots, x_m)$
- $TS_2 = (y_1, y_2, \ldots, y_n)$ (có thể $m \neq n$)

. Dynamic Time Warping (DTW)

- **Mô tả:** Thuật toán tìm sự tương ứng tối ưu giữa hai chuỗi thời gian có thể khác nhau về tốc độ hoặc độ dài bằng cách "biến dạng" trục thời gian.
- **Công thức (khoảng cách DTW tích lũy):**

$$D(i,j) = \text{dist}(x_i, y_j) + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

  - $\text{dist}(x_i, y_j)$: khoảng cách giữa hai điểm (thường là Euclid).
  - $D(i,j)$: khoảng cách DTW tích lũy nhỏ nhất đến điểm (x_i, y_j).
- **Ý nghĩa:** Cho phép so sánh các chuỗi có độ dài và biến thiên thời gian khác nhau, hữu ích trong nhận dạng giọng nói, phân tích chuyển động. Khoảng cách DTW càng nhỏ, hai chuỗi càng tương đồng.

VD

- GS có hai chuỗi thời gian đơn giản này:
  - TS1=(1,3,4) (có 3 điểm)
  - TS2=(2,3,5) (có 3 điểm)
- Mục tiêu của DTW là tìm ra con đường "ghép nối" các điểm từ TS1 với các điểm từ TS2 sao cho tổng khoảng cách tích lũy là nhỏ nhất. Con đường này không nhất thiết phải thẳng hàng (1 với 2, 3 với 3, 4 với 5) mà có thể "co giãn" theo thời gian.

3.4 Graph Similarity Measures

- Ứng dụng: mạng xã hội, web, protein...
- Độ đo:
- - Graph edit distance
- - Subgraph isomorphism
- - Graph kernel methods
- - Jaccard (trên cạnh/đỉnh)

Graph Edit Distance (GED)

1. Graph Edit Distance (GED)

- **Mô tả:** Khoảng cách được định nghĩa là chi phí tối thiểu để biến đổi một đồ thị thành đồ thị khác thông qua một chuỗi các thao tác chỉnh sửa (thêm/xóa nút, thêm/xóa cạnh, thay đổi nhãn).
- **Công thức:**

$$GED(G_1, G_2) = \min_{(e_1,\ldots,e_k)\in P(G_1,G_2)} \sum_{i=1}^{k} c(e_i)$$

  - $P(G_1, G_2)$: tập hợp chuỗi thao tác chỉnh sửa.
  - e_i: một thao tác chỉnh sửa.
  - $c(e_i)$: chi phí của thao tác e_i.
- **Ý nghĩa:** Cung cấp phép đo khoảng cách "ngữ nghĩa" giữa các đồ thị. Đây là bài toán NP-hard.

vd

- **Ví dụ Minh Họa:**
- Chi phí: Thêm/Xóa nút = 1; Thêm/Xóa cạnh = 0.5; Đổi nhãn nút = 1.
- G1: A--B
- G2: A--C
- **Một chuỗi thao tác biến G1 thành G2:**
  - Xóa cạnh B (giữa A và B): chi phí 0.5
  - Đổi nhãn nút B thành C: chi phí 1
  - Thêm cạnh A--C: chi phí 0.5
  - **Tổng chi phí (một khả năng):** 0.5+1+0.5=2

Tổng kết độ đo

- Loại dữ liệu | Độ đo chính
- ----------|----------
- Đa chiều | Euclidean, Manhattan
- Văn bản | Cosine, Jaccard, Edit
- Thời gian | DTW, Euclidean
- Đồ thị | Edit distance, Kernel

Bài tập cuối chương

- 1. Tính cosine similarity giữa 2 văn bản bất kỳ.
- 2. So sánh 2 chuỗi thời gian ngắn bằng Euclidean và DTW.
- 3. Cho 2 đồ thị nhỏ (3-5 đỉnh), tính graph edit distance.
- 4. Giải thích ứng dụng thực tế của các độ đo.

Diễn giải nâng cao

- - Độ đo ảnh hưởng đến kết quả phân cụm, phân loại
- - Chọn độ đo phù hợp với dữ liệu
- - SVM cần định nghĩa rõ similarity function

Kết thúc

- Câu hỏi? Thảo luận!
- Tài liệu: "Data Mining: The Textbook" – Charu C. Aggarwal
- [Tên Giảng viên, Bộ môn, Ngày trình bày]