**CHƯƠNG 2**

**1. biểu diễn các văn bản đó dưới dạng vector trọng số theo kỹ thuật bảng tần số**

**Ví dụ: xét 1 tập văn bản được ký hiệu sau d1,d2,d3,d4,d5,d6. Với tần suất xuất hiện của các từ t1,t2,t3,t4,t5 trong các văn bản trình bày như sau:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Term/Document** | **d1** | **d2** | **d3** | **d4** | **d5** | **d6** |
| **t1** | 615 | 390 | 10 | 10 | 18 | 65 |
| **t2** | 15 | 4 | 76 | 217 | 91 | 816 |
| **t3** | 2 | 8 | 815 | 142 | 765 | 1 |
| **t4** | 312 | 511 | 677 | 11 | 711 | 2 |
| **t5** | 45 | 33 | 516 | 64 | 491 | 59 |

*Bảng 2.2. Bảng tần suất các từ trong các văn bản*

* Mỗi văn bản dj được biểu diễn bởi 1 vector chỉ tần suất xuất hiện của các từ trong văn bản đó : (tf1,j, tf2,j, … ,tfM,j). Thông thường các tần suất này được chuẩn hóa về [0, 1]: để tính đến ảnh hưởng của độ dài của văn bản. Nếu tần suất quá nhỏ ta có thể đưa về giá trị 0.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Term/Document** | **d1** | **d2** | **d3** | **d4** | **d5** | **d6** |
| **t1** | 0.62 | 0.41 | 0.00 | 0.02 | 0.01 | 0.07 |
| **t2** | 0.02 | 0.00 | 0.04 | 0.49 | 0.04 | 0.87 |
| **t3** | 0.00 | 0.01 | 0.39 | 0.32 | 0.37 | 0.00 |
| **t4** | 0.32 | 0.54 | 0.32 | 0.02 | 0.34 | 0.00 |
| **t5** | 0.05 | 0.03 | 0.25 | 0.14 | 0.24 | 0.06 |

*Bảng 2.3. Bảng tần suất các từ trong các văn bản sau khi chuẩn hóa*

* idf (inverse document frequency): xác định độ quan trọng của mỗi từ trong tập dữ liệu văn bản đang xem xét. Ở đây độ quan trọng của mỗi từ được xác định theo công thức:

idfi = log(N/dfi).

* + Ở đây, N: là tổng số văn bản trong tập dữ liệu, dfi: là số văn bản có chứa từ ti.
  + Như vậy, trọng số tf.idf của từ ti trong văn bản dj là: wi,j = tf(i,j) x idf(i).
  + Mỗi văn bản dj được biểu diễn bởi 1 vector tf.idf: (w1,j, w2,j, … , wM,j)

Với ví dụ trên, ta có thể tính được độ quan trọng của từ t1 trong tập văn vản đang xem xét: idf1 = log(6/5) = 0.079. Tương tự ta có thể tính được độ quan trọng của từ t2,t3,t4,t5 trong tập văn bản:

idf2 = idf4 = log(6/5)=0.079, idf5 = log(6/6) = 0, idf3 = log(6/4)=0.176

Từ đó ta tính được trọng số của từ t1 trong các văn bản dj là:

w1,1 = 0.079 \* 0.62 = 0.04898. w12 = 0.079\*0.41 = 0.03239.

- Tương tự ta tính được:

w13=0.079\*0=0,w14=0.079\*0.02=0.00158,w15=0.079\*0.01=0.00079, w16=0.079\*0.07=0.00553.

- Tương tự:

w21=0.079\*0.02=0.00158,w22=0,w23=0.079\*0.04=0.00316,w24=0.079\*0.49=0.03871,w25=0.079\*0.04=0.00316,w26=0.079\*0.87=0.06873

w31=0,w32=0.176\*0.01=0.00176,w33=0.176\*0.39=0.06864,w34=0.176\*0.32=

0.05632, w35=0.176\*0.37=0.06512,w36=0.

w41=0.079\*0.32=0.02528,w42=0.079\*0.54=0.04266,w43=0.079\*0.32=0.02528,w44=0.079\*0.02=0.00158,w45=0.079\*0.34=0.02686,w46=0

w51=0,w52=0,w53=0,w54=0,w55=0,w56=0

Như vậy mỗi văn bản được biểu diễn bởi 1 vector:

d1(w11,w21,w31,w41,w51)=(0.04898,0.00158,0,0.02528,0)

tương tự: d2,d3,d4,d5,d6.

d2(w12,w22,w32,w42,w52)

d3(w13,w23,w33,w43,w53)

d4(w14,w24,w34,w44,w54)

d5(w15,w25,w35,w45,w55)

d6(w16,w26,w36,w46,w56)

***Bài tập 1:* cho các văn bản và tần suất xuất hiện các từ trong văn bản. Hãy biểu diễn các văn bản đó dưới dạng vector trọng số theo kỹ thuật bảng tần số:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| term/doc | d1 | d2 | d3 | d4 | d5 | d6 | d7 |
| t1 | 0.03 | 0 | 0 | 0.04 | 0.31 | 0.22 | 0 |
| t2 | 0.64 | 0.33 | 0.01 | 0 | 0 | 0 | 0.05 |
| t3 | 0 | 0 | 0.25 | 0.29 | 0.01 | 0.01 | 0.05 |
| t4 | 0 | 0.6 | 0.33 | 0.27 | 0 | 0.12 | 0.3 |
| t5 | 0.55 | 0 | 0 | 0 | 0.12 | 0 | 0.01 |
| t6 | 0.47 | 0.02 | 0 | 0 | 0.15 | 0.56 | 0.2 |
| t7 | 0.01 | 0.13 | 0.05 | 0 | 0 | 0 | 0 |

**2. Truy xuất thông tin văn bản**

Sau khi đánh chỉ mục cho văn bản (Indexing), tiếp theo là quá trình tìm kiếm thông tin văn bản (Retrieving/Searching). Đây là quá trình tìm kiếm, đối sánh thông tin trên tập các văn bản đã đánh chỉ mục làm sao cho việc truy vấn hiệu quả. Quá trình truy vấn diễn ra theo quy trình:

* Biểu diễn các tài liệu văn bản Di dưới 1 tập các chỉ mục. Biểu diễn văn bản Q cần tìm kiếm cùng với dạng biểu diễn của tập văn bản đó.
* Sau khi biểu diễn xong, ta thực hiện việc đối sánh Q với tập dữ liệu văn bản đó trong Cơ sở dữ liệu văn bản. Tức là xác định khoảng cách giữa Q và các Di.

Có nhiều phương pháp truy vấn thông tin văn bản, sau đây chúng ta sẽ xem xét tới phương pháp Vector, đây là 1 trong những phương pháp thông dụng nhất trong truy vấn văn bản.

1. **Phương pháp Vector:**
   * Giả sử các văn bản và truy vấn đều được biểu diễn bởi 1 tập cố định M khái niệm/từ (term) có trọng số. Mỗi văn bản Dj, truy vấn Qi được biểu diễn bởi 1 vector:
   * Ở đây: wjk, wik: là các trọng số của từ k trong Dj và Qi. wlk: {0, 1}. Tức là các văn bản được biểu diễn dưới dạng các vector trọng số với tập cố định M terms.
   * **Tính khoảng cách giữa 2 vector: Dj và Qi. Có 2 phương pháp tính khoảng cách đó là khoảng cách Euclide và khoảng cách Cosine.**

* Khoảng cách Euclide
* **Khoảng cách Cosine: 1 - S(Qi,Dj)** 
  + Kết quả thu được sẽ được sắp xếp (ranking) theo thứ tự giảm dần của độ tương tự S(Q,D), hay tăng dần của khoảng cách cosine (1-S(Q,D)).
  + Ví dụ cho 1 tập các văn bản D được biểu diễn dưới dạng các vector có trọng số. Hãy sắp xếp theo thứ tự giảm dần độ tương tự của các văn bản D với văn bản cần tìm Q. Thông thường hay sử dụng theo phương pháp đo khoảng cách Cosine:

D1 = [0.2, 0.1, 0.4, 0.5]

D2 = [0.5, 0.6, 0.3, 0]

D3 = [0.4, 0.5, 0.8, 0.3]

D4 = [0.1, 0, 0.7, 0.8]

Q = [0.5, 0.5, 0, 0]

Ta tính được:

Tương tự S(D2,Q) = 0.93, S(D3,Q) = 0.66, S(D4,Q) = 0.07.

* **Kết quả sắp xếp: D2, D3, D1, D4.**

– Ưu điểm của phương pháp tìm kiếm Vector là cho phép tìm kiếm ra kết quả gần đúng (partial matching), đo được mức độ giống nhau giữa văn bản và truy vấn. Và phương pháp này tương đối đơn giản, thích hợp với các văn bản ngắn.

- Nhược điểm của phương pháp tìm kiếm Vector là coi các term không có liên quan với nhau cho nên chưa xác định được mối liên quan không gian giữa các từ. Và phương pháp này độ phức tạp khi tìm kiếm: O(M x N) lớn khi M, N lớn, M: số từ trong từ điển (tiếng anh >10 000 000 từ).

*Bài tập 2.* cho các văn bản Di và tần suất xuất hiện các từ tj trong văn bản, và văn bản Q là văn bản cần tìm. Hãy biểu diễn các văn bản đó dưới dạng vector trọng số theo kỹ thuật bảng tần số. Sau đó hãy sắp xếp theo thứ tự giảm dần độ tương tự của các văn bản D với văn bản cần tìm Q theo phương pháp đo khoảng cách Cosine.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| term/doc | d1 | d2 | d3 | d4 | d5 | d6 | **Q** |
| t1 | 0.03 | 0 | 0 | 0.04 | 0.31 | 0.22 | **0** |
| t2 | 0.64 | 0.33 | 0.01 | 0 | 0 | 0 | **0.05** |
| t3 | 0 | 0 | 0.25 | 0.29 | 0.01 | 0.01 | **0.05** |
| t4 | 0 | 0.6 | 0.33 | 0.27 | 0 | 0.12 | **0.3** |
| t5 | 0.55 | 0 | 0 | 0 | 0.12 | 0 | **0.01** |
| t6 | 0.47 | 0.02 | 0 | 0 | 0.15 | 0.56 | **0.2** |
| t7 | 0.01 | 0.13 | 0.05 | 0 | 0 | 0 | **0** |

***Hướng dẫn giải:***

* Bước1: Tính độ quan trọng của mỗi từ idfi, với i=1-7 (với số văn bản xét N=6).
* Bước 2: Các văn bản di được biểu diễn bởi 1 vector. Văn bản Q cũng được biểu diễn dưới 1 vector.
* Bước 3: sử dụng phương pháp đo khoảng cách cosine để đo khoảng cách từ Q tới các di
* Bước 4: dựa trên các khoảng cách ta có thể sắp xếp thứ tự giảm dần độ tương tự giữa Q với các văn bản di bằng cách sắp xếp khoảng cách theo thứ tự tăng dần.

***Bài tập 3.*** Cho các văn bản D1,D2,D3,D4 và văn bản Q với tần suất xuất hiện của các từ trong văn bản tương ứng, hãy sắp xếp theo độ giống nhau giảm dần của các văn bản D1,D2,D3,D4 so với văn bản Q.

D1(0.1,0.3,0.2,0.4,0.5)

D2(0.3,0.2,0.4,0.3,0.1)

D3(0.4,0.1,0.2,0.4,0.3)

D4(0.3,0.5,0.2,0.1,0.1)

Q(0.2,0.4,0.1,0.1,0.3)

**3.** **Thuật toán tìm kiếm chuỗi gần đúng**

Sự tương tự giữa hai chuỗi ký tự theo các độ đo kinh điển (dãy con chung dài nhất, xâu con chung dài nhất, khoảng cách chỉnh sửa - Edit, …) chỉ hiệu quả khi có lỗi xảy ra ở mẫu hay văn bản do một số ký tự bị xóa, chèn, thay thế và đều coi trọng trật tự của các ký tự trong chuỗi. Tuy nhiên các tiếp cận này chưa đáp ứng đầy đủ nhu cầu thực tế khi tìm kiếm thông tin tương tự trong các hệ thống xử lý văn bản.

Khi ứng dụng trong thực tế, mỗi từ có thể xem là một “ký tự hình thức”. Với mẫu P là câu “Hà Nội là thủ đô của nước Việt Nam”, có thể hình thức hóa P = abcdefghk và chuỗi S là “Đất nước Việt Nam chúng ta có thủ đô là Hà Nội”, và hình thức hóa S = lghkmnodecab. Tính độ tương tự giữa hai chuỗi theo các độ đo kinh điển được các kết quả như sau:

* Độ dài dãy con chung dài nhất: 3, vậy độ tương tự có thể xem là 3/12.
* Khoảng cách Edit (phép biến đổi): 15 (số phép xóa chèn hay thay thế ký tự để biến đổi chuỗi P thành S).
* Khoảng cách Hamming chỉ có thể áp dụng cho hai chuỗi có độ dài bằng nhau nên không thể áp dụng để tính độ tương tự giữa hai chuỗi này.

Bằng các phương pháp trên, độ tương tự giữa P và S rất nhỏ, song về mặt ngữ nghĩa có thể nói là rất gần nhau. Các tình huống tương tự rất hay gặp trong thực tế, như khi cần tìm kiếm tên người nước ngoài (chẳng hạn ‘C.J.Date’ và ‘Date.C.J’, “Christian Charras” và “Charas C.”), khi có sự sai khác do biến đổi hình thái từ, cấu trúc câu (“approximate searching” và “search approximately”), một số trường hợp thứ tự ghép từ khác nhau nhưng mang ngữ nghĩa gần giống nhau (“toán logic” và “logic toán”, “lung linh” và “linh lung”) hoặc do thứ tự sai song vẫn hiểu được đúng nghĩa (“toán giải tích” và “giải tích toán”).

Độ tương tự được đề xuất ở đây cho phép đo độ tương tự về mặt hình thức giữa hai chuỗi theo quan điểm thống kê.

Trong các [thuật toán](http://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) của bộ môn khoa học máy tính, khái niệm Khoảng cách Levenshtein thể hiện khoảng cách khác biệt giữa 2 chuỗi kí tự. Khoảng cách Levenshtein giữa chuỗi S và chuỗi T là số bước ít nhất biến chuỗi S thành chuỗi T thông qua 3 phép biến đổi là:

* Xoá 1 kí tự.
* Thêm 1 kí tự.
* Thay kí tự này bằng kí tự khác.

Khoảng cách này được đặt theo tên [Vladimir Levenshtein](http://vi.wikipedia.org/wiki/Vladimir_Iosifovich_Levenshtein), người đã đề ra khái niệm này vào năm 1965. Nó được sử dụng trong việc tính toán sự giống và khác nhau giữa 2 chuỗi, như chương trình kiểm tra lỗi chính tả của [winword](http://vi.wikipedia.org/w/index.php?title=Winword&action=edit&redlink=1) [spellchecker](http://vi.wikipedia.org/w/index.php?title=Spellchecker&action=edit&redlink=1). Ví dụ: Khoảng cách Levenshtein giữa 2 chuỗi "kitten" và "sitting" là 3, vì phải dùng ít nhất 3 lần biến đổi.

1. kitten -> sitten (thay "k" bằng "s")
2. sitten -> sittin (thay "e" bằng "i")
3. sittin -> sitting (thêm kí tự "g")

Để tính toán Khoảng cách Levenshtein, ta sử dụng thuật toán [quy hoạch động](http://vi.wikipedia.org/wiki/Quy_ho%E1%BA%A1ch_%C4%91%E1%BB%99ng), tính toán trên mảng 2 chiều (n+1)\*(m+1), với n, m là độ dài của chuỗi cần tính. Sau đây là đoạn mã (S, T là chuỗi cần tính khoảng cách, n, m là độ dài của chuỗi S, T):

int LevenshteinDistance(char s[1..m], char t[1..n])

// d là 1 bảng với m+1 hàng và n+1 cột

{ declare int d[0..m, 0..n]

for i from 0 to m

d[i, 0] := i

for j from 0 to n

d[0, j] := j

for i from 1 to m

for j from 1 to n

{

if s[i] = t[j] then cost := 0

else cost := 1

d[i, j] := minimum(

d[i-1, j] + 1, // trường hợp xoá

d[i, j-1] + 1, // trường hợp thêm

d[i-1, j-1] + cost // trường hợp thay thế

)

}

return d[m, n]

}

Ví dụ1.  Giá trị bảng so sánh chuối kí tự

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **s** | **a** | **t** | **u** | **r** | **d** | **a** | **y** |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **S** | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **U** | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 5 | 6 |
| **N** | 3 | 2 | 2 | 2 | 3 | 3 | 4 | 5 | 6 |
| **D** | 4 | 3 | 3 | 3 | 3 | 4 | 3 | 4 | 5 |
| **A** | 5 | 4 | 3 | 4 | 4 | 4 | 4 | 3 | 4 |
| **Y** | 6 | 5 | 4 | 4 | 5 | 5 | 5 | 4 | **3** |

*Bảng 2.6. bảng kết quả so sánh chuỗi theo phương pháp V.L*

***Bài tập:*** Hãy so sánh độ dài các chuỗi sau:

* “Bill clinton” với “clinton Bill”
* “Ha noi viet nam” với “Viet nam ha loi”

### 4. Truy vấn ảnh dựa trên đặc trưng màu sắc

Các bước chính của kiểu truy vấn dựa trên nội dung ảnh đó là trích chọn đặc trưng (Biểu diễn ảnh – Feature extraction), đánh chỉ mục để tăng hiệu năng tìm kiếm (indexing) và lựa chọn thước đo độ tương tự.

Trước tiên chúng ta tìm hiểu về cách biểu diễn ảnh – trích chọn đặc trưng ảnh, có 2 loại đặc trưng ảnh:

* Đặc trưng toàn cục: là các đặc trưng được trích chọn trên toàn bộ bức ảnh, đó là màu sắc, hình dáng, kết cấu, … của cả bức ảnh.
* Đặc trưng về màu sắc: Histogram là 1 đặc trưng, Histogram thể hiện sự phân bố màu sắc trên ảnh. Cách tính Histogram đã được trình bày trong môn Xử lý ảnh, đó là:

Bước 1: Lấy mẫu màu của ảnh. VD, ảnh màu 24 bit : 8 x 8 x8 màu (bin)

Bước 2: Tính số điểm ảnh tương ứng với mỗi màu.

Bước 3: Chuẩn hóa: chia cho tổng số điểm ảnh. Ta có lược đồ Histogram H của ảnh:

H = (h1, h2, . . . , hn), hi ϵ [0,1], Ʃhi = 1

Khoảng cách giữa 2 ảnh dựa trên phân bố màu có thể được tính toán theo 2 công thức sau:

* Khoảng cách euclide: Khoảng cách giữa 2 bức ảnh H và G
* Giao giữa 2 Histogram: Khoảng cách giữa 2 ảnh H và G

Thường hay sử dụng theo công thức giao giữa 2 Histogram để so sánh khoảng cách giữa 2 ảnh dựa trên phân bố màu.

Ưu điểm của phân bố màu đó là không phụ thuộc/ ít phụ thuộc vào một số các biến đổi hình học: Phép quay, phép tịnh tiến, phóng to, thu nhỏ,... bởi vì khi ảnh bị biến đổi hình học thì phân bố màu của nó không thay đổi.



*Hình 2.9. Các phép biến đổi hình học không ảnh hưởng tới lược đồ màu*

***Bài tập ví dụ:*** cho các ảnh đa mức xám dưới đây X1,X2,X3 với mức xám trong khoảng [0-7], và 1 ảnh đa mức xám cần truy vấn Q. Hãy sắp xếp theo thứ tự mức độ gần giống Q nhất của các ảnh trên. (ảnh thông thường có mức xám trong khoảng 0-255).

|  |  |  |  |
| --- | --- | --- | --- |
| X1 |  |  |  |
|  | 1 | 3 | 3 |
|  | 0 | 6 | 2 |
|  | 5 | 4 | 5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X2 |  |  |  |  |
|  | 0 | 0 | 1 | 2 |
|  | 2 | 2 | 0 | 2 |
|  | 1 | 2 | 1 | 6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X3 |  |  |  |  |
|  | 1 | 3 | 4 | 4 |
|  | 0 | 2 | 3 | 6 |
|  | 2 | 2 | 7 | 6 |
|  | 3 | 2 | 5 | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| Q |  |  |  |
|  | 2 | 3 | 2 |
|  | 1 | 2 | 0 |
|  | 4 | 3 | 1 |

***Hướng dẫn giải:***

Tính biểu đồ histogram của ảnh X1,X2,X3,Q:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **X1** |  |  |  |  |  |  |  |  |
| Mức xám | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| n1i | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 0 |
| h1i | 1/9 | 1/9 | 1/9 | 2/9 | 1/9 | 2/9 | 1/9 | 0 |
| **X2** |  |  |  |  |  |  |  |  |
| n2i | 3 | 3 | 5 | 0 | 0 | 0 | 1 | 0 |
| h2i | 3/12 | 3/12 | 5/12 | 0 | 0 | 0 | 1/12 | 0 |
| **X3** |  |  |  |  |  |  |  |  |
| n3i | 1 | 1 | 4 | 3 | 2 | 1 | 3 | 1 |
| h3i | 1/16 | 1/16 | 4/16 | 3/16 | 2/16 | 1/16 | 3/16 | 1/16 |
| **Q** |  |  |  |  |  |  |  |  |
| nQi | 1 | 2 | 3 | 2 | 1 | 0 | 0 | 0 |
| hQi | 1/9 | 2/9 | 3/9 | 2/9 | 1/9 | 0 | 0 | 0 |

Tính khoảng cách giữa Xi tới Q theo phương pháp khoảng cách euclide:

d(X1,Q)=0+1/9+2/9+0+0+2/9+1/9+0=0.667

d(X2,Q)=|3/12-1/9|+|3/12-2/9|+|5/12-3/9|+2/9+1/9+0+1/12+0=0.667

d(X3,Q)=|1/16-1/9|+|1/16-2/9|+|4/16-3/9|+|3/16-2/9|+|2/16-1/9| +1/16 +3/16 +1/16 = 0.576

Như vậy Q gần giống X3 nhất, tiếp theo là X1, X2.

***Bài tập 2.*** Cho các ảnh đa mức xám dưới đây X1,X2,X3 với mức xám trong khoảng [0-8], và 1 ảnh đa mức xám cần truy vấn Q. Hãy sắp xếp theo thứ tự mức độ gần giống Q nhất của các ảnh trên. (ảnh thông thường có mức xám trong khoảng 0-255).

|  |  |  |  |
| --- | --- | --- | --- |
| X1 |  |  |  |
|  | 1 | 3 | 3 |
|  | 0 | 8 | 2 |
|  | 5 | 4 | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| X3 |  |  |  |
|  | 2 | 3 | 2 |
|  | 1 | 7 | 0 |
|  | 4 | 3 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q |  |  |  |  |
|  | 1 | 3 | 4 | 4 |
|  | 0 | 2 | 3 | 6 |
|  | 8 | 2 | 7 | 6 |
|  | 3 | 2 | 5 | 6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X2 |  |  |  |  |
|  | 0 | 0 | 1 | 2 |
|  | 2 | 2 | 8 | 2 |
|  | 1 | 2 | 1 | 6 |

### B.1. Phân vùng ảnh

### 

### 5. Phân vùng bằng ngưỡng

Một trong những phương pháp đơn giản nhất thường được sử dụng cho phân đoạn các đối tượng trong ảnh đó là phân đoạn bằng ngưỡng (Threshold).

Ngưỡng là 1 giá trị cụ thể trong khoảng [0-255] dùng để phân vùng cho ảnh. Ngưỡng có thể là ngưỡng toàn cục: một ngưỡng cho toàn bộ ảnh, hoặc ngưỡng Cục bộ: một ngưỡng cho 1 vùng của ảnh. Có thể dùng n ngưỡng để phân ảnh thành (n+1) vùng. Ngưỡng thường được lựa chọn phù hợp với từng ảnh hoặc từng vùng của ảnh.

Ví dụ chia ảnh thành 2 vùng dựa trên ngưỡng toàn cục:

– IF value(pixel) >= threshold THEN value(pixel) = 1

– IF value(pixel) < threshold THEN value(pixel) = 0

Đây là chuyển ảnh thành Ảnh nhị phân.

Ví dụ về đa ngưỡng - Sử dụng n ngưỡng phân ảnh thành n+1 vùng, ở đây sử dụng ngưỡng cục bộ, 1 ngưỡng cho 1 vùng:

- IF value(pixel) < threshold\_1 THEN value(pixel) = 0

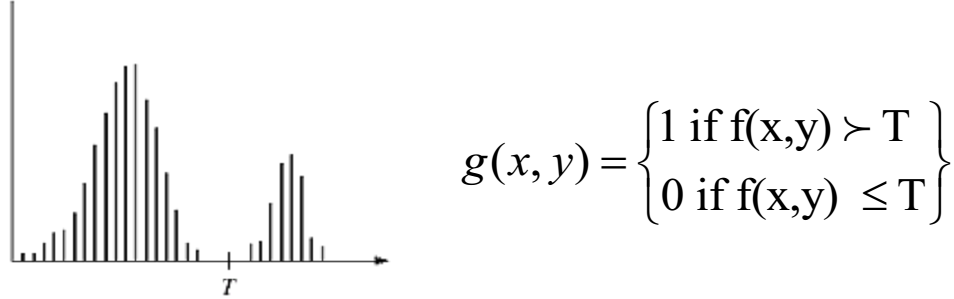
- IF value(pixel) >= threshold\_1 && value(pixel) < threshold\_2 THEN value(pixel) = 1

- …

- IF value(pixel) >= threshold\_n THEN value(pixel) = n

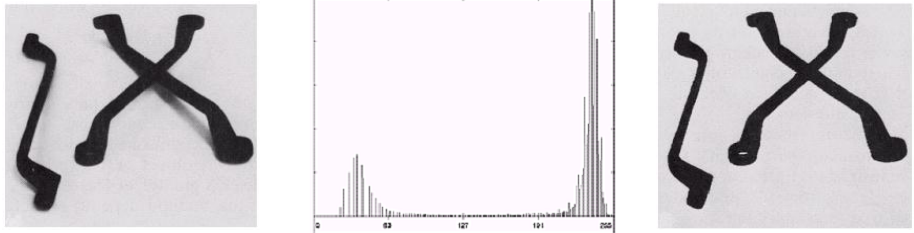
Vấn đề chính ở đây là lựa chọn giá trị ngưỡng thế nào và số ngưỡng xác định thế nào?

Lựa chon ngưỡng toàn cục dựa trên phân bố là 1 lựa chọn đơn giản nhất. Phù hợp cho môi trường ổn định, thường được ứng dụng công nghiệp.



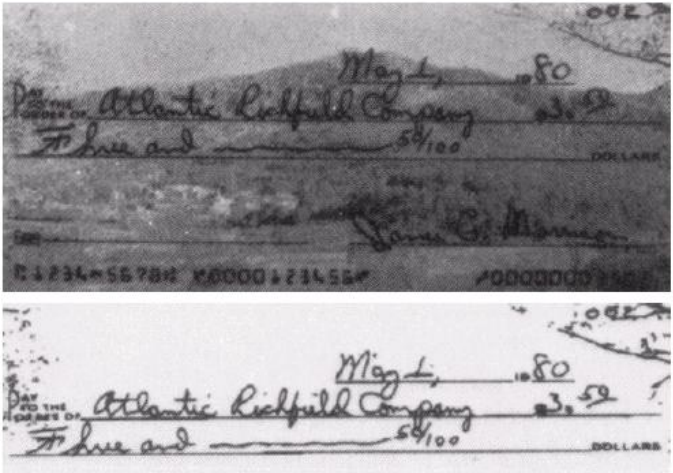
*Hình 2.17. Ví dụ về lựa chọn ngưỡng phân vùng*

Ví dụ cho ảnh sau, xác định ngưỡng và phân vùng ảnh dựa trên ngưỡng toàn cục, kết quả ta được 1 ảnh nhị phân:



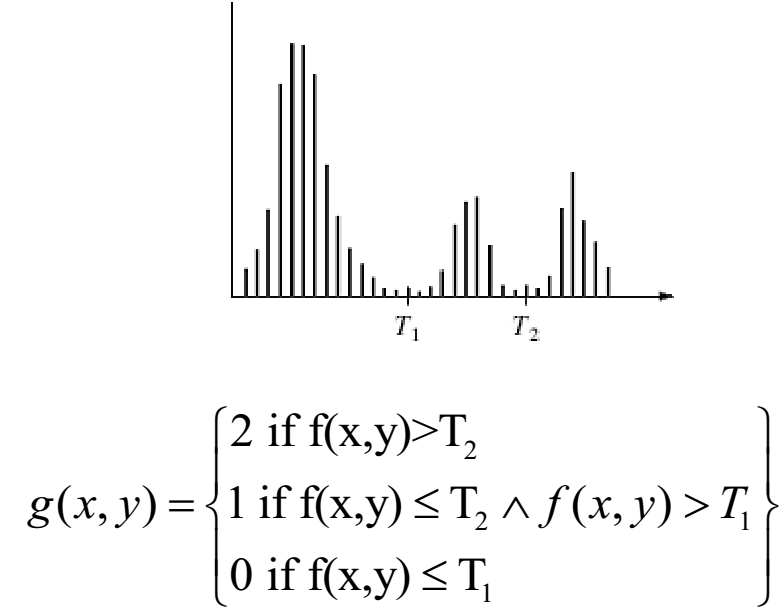
*Hình 2.18. Ví dụ về ảnh trước và sau khi phân vùng theo ngưỡng.*

***Bài tập áp dụng:*** Hãy xây dựng 1 phần mềm, với đầu vào là 1 ảnh đa mức xám (Hoặc ảnh màu, sau đó chuyển về ảnh đa mức xám), hãy xây dựng biểu đồ histogram cho ảnh, sau đó xác định ngưỡng toàn cục cho ảnh trên. Tiếp theo hãy phân vùng ảnh với ngưỡng toàn cục đã được xác định. Đầu ra là 1 ảnh nhị phân. Mô phỏng kết quả như trên hình.



*Hình 2.19. Ảnh minh họa trước và sau phân vùng theo ngưỡng*

Trong ngưỡng cục bộ, bài toán đặt ra là số ngưỡng xác định thế nào? một cách đơn giản nhất đó là xác định đa ngưỡng dựa trên phân bố màu của ảnh (Histogram).



*Hình 2.20. Ví dụ về đa ngưỡng*

Sau khi xác định được các ngưỡng, chúng ta có thể phân vùng ảnh.

***Bài tập ví dụ:*** Cho 1 ảnh đa mức xám sau, hãy xác định số ngưỡng, giá trị ngưỡng theo phân bố màu histogram, sau đó hãy phân vùng ảnh theo số ngưỡng và giá trị ngưỡng vừa xác định.

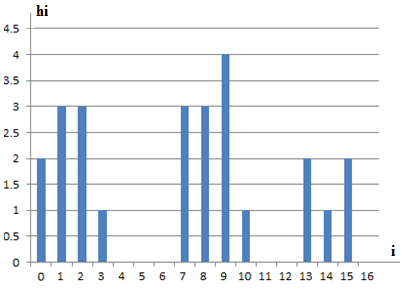
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X |  |  |  |  |  |
|  | 0 | 0 | 1 | 8 | 8 |
|  | 2 | 1 | 2 | 9 | 9 |
|  | 1 | 2 | 9 | 7 | 13 |
|  | 3 | 9 | 10 | 13 | 15 |
|  | 7 | 8 | 7 | 14 | 15 |

***Hướng dẫn giải:***

Trước tiên ta xác định phân bố màu cho ảnh trên:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Mức xám | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| ni | 2 | 3 | 3 | 1 | 0 | 0 | 0 | 3 | 3 | 4 | 1 | 0 | 0 | 2 | 1 | 2 | 0 |
| hi | 2/25 | 3/25 | 3/25 | 1/25 | 0 | 0 | 0 | 3/25 | 3/25 | 4/25 | 1/25 | 0 | 0 | 2/25 | 1/25 | 2/25 | 0 |

Vẽ biểu đồ histogram cho ảnh:



Dựa trên biểu đồ histogram, ta xác định (định tính) ảnh chia làm 3 vùng với 3 ngưỡng có giá trị T1=6, T2=12, T3=16. Ảnh kết quả sau khi phân ngưỡng là:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X |  |  |  |  |  |
|  | 0 | 0 | 0 | 1 | 1 |
|  | 0 | 0 | 0 | 1 | 1 |
|  | 0 | 0 | 1 | 1 | 2 |
|  | 0 | 1 | 1 | 2 | 2 |
|  | 1 | 1 | 1 | 2 | 2 |

***Bài tập 2.*** Cho 1 ảnh đa mức xám sau, hãy xác định số ngưỡng, giá trị ngưỡng theo phân bố màu histogram, sau đó hãy phân vùng ảnh theo số ngưỡng và giá trị ngưỡng vừa xác định.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X |  |  |  |  |  |
|  | 0 | 0 | 1 | 8 | 8 |
|  | 2 | 1 | 2 | 9 | 9 |
|  | 1 | 2 | 9 | 7 | 12 |
|  | 3 | 9 | 10 | 11 | 10 |
|  | 7 | 8 | 8 | 11 | 10 |

### 

### 6. Phân vùng bằng giải thuật phân cụm K-means

Một phương pháp thường dùng để phân đoạn ảnh thành nhiều vùng (K vùng), đó là sử dụng giải thuật K-Means trong phân nhóm (Clusters). Ở đây các điểm ảnh trong ảnh được phân vào K nhóm, K biết trước và cố định, K là số nhóm mà bức ảnh cần phân nhóm.

*Vậy Phân cụm là gì?* là quá trình phân chia 1 tập dữ liệu ban đầu (n điểm ảnh) thành các cụm dữ liệu thỏa mãn các đối tượng trong 1 cụm “tương tự” nhau và các đối tượng khác cụm thì “không tương tự” nhau về tính chất.

Tư tưởng của phân cụm là phân 1 tập dữ liệu có n phần tử cho trước thành k tập con dữ liệu (k ≤ n), mỗi tập con biểu diễn 1 cụm. Các cụm hình thành trên cơ sở làm tối ưu giá trị hàm đo độ tương tự sao cho: Các đối tượng trong 1 cụm là tương tự và Các đối tượng trong các cụm khác nhau là không tương tự nhau. Ở đây mỗi đối tượng chỉ thuộc về 1 cụm và mỗi cụm có tối thiểu 1 đối tượng.

**Giải thuật K-means:**

Đầu vào: - Tập các đối tượng X = {xi| i = 1, 2, …, N},

- Số cụm: K

Đầu ra: - Các cụm Ci ( i = 1 ÷ K) tách rời.

-Bước 1 - Khởi tạo

Chọn K trọng tâm ứng với K đối tượng bất kỳ {ci} (i = 1÷K).

- Bước 2 - Tính toán khoảng cách

Si(t)= {Xj : ||Xj - ci(t) || ≤ ||Xj - ci\*(t) || với tất cả i = 1, …, k}

- Bước 3 - Cập nhật lại trọng tâm

- Bước 4 – Điều kiện dừng

Lặp lại các bước 2 và 3 cho tới khi không có sự thay đổi trọng tâm của cụm.

+ Các phép đo khoảng cách thường được dùng trong gom cụm K-Means:

* + ***Minkowski***

Với xi,yi là 2 vector

* + ***Euclidean:*** Độ đo Minkowski trong trường hợp p = 2

***Ví dụ minh họa:*** Cho 4 điểm ảnh có tọa độ (X,Y) dưới đây. Hãy phân các điểm này thành k=2 cụm.

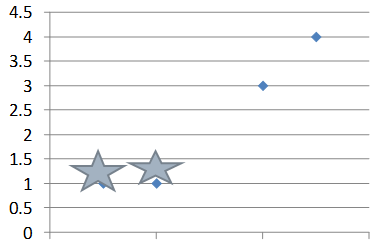
|  |  |  |
| --- | --- | --- |
| **Objects** | **X** | **Y** |
| A | 1 | 1 |
| B | 2 | 1 |
| C | 4 | 3 |
| D | 5 | 4 |

*Hình 2.21. Các đối tượng ban đầu*

**- Bước 1:** Khởi tạo

Chọn 2 trọng tâm ban đầu:

c1(1,1) ≡ A và c2(2,1) ≡ B, thuộc 2 cụm 1 và 2



*Hình 2.22. Chọn các trọng tâm ngẫu nhiên ban đầu*

**- Bước 2:** Tính toán khoảng cách (bình phương)

d(C, c1) = (4-1)2 + (3-1)2 = 13

d(C, c2) = (4-2)2 + (3-1)2 = 8

So sánh d(C, c1) > d(C, c2) => phân C thuộc cụm 2

d(D, c1) = (5-1)2 + (4-1)2  = 25

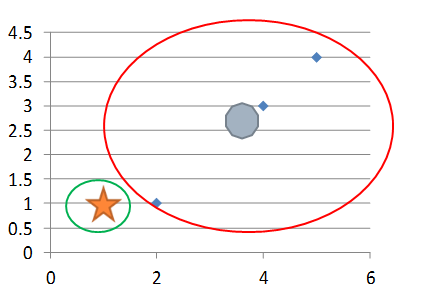
d(D, c2) = (5-2)2 + (4-1)2 = 18

So sánh d(D,c1) > d(D, c2) => phân D thuộc cụm 2

Sau bước 2 ta phân ra các cụm: cụm 1{A}, cụm 2{B,C,D}

**- Bước 3:** Cập nhật lại vị trí trọng tâm

Trọng tâm cụm 1: c1 ≡ A (1, 1), Trọng tâm cụm 2: c2(x,y) =((2+4+5)/3,(1+3+4)/3)



*Hình 2.23. Các cluster sau phân hoạch lần 1*

Do trọng tâm có sự thay đổi, cho nên ta lặp lại bước 2.

**- Bước 4-1:** Lặp lại bước 2 – Tính toán khoảng cách

d(A, c1 ) = 0 < d(A, c2 ) = 9.89 => Phân A thuộc cụm 1

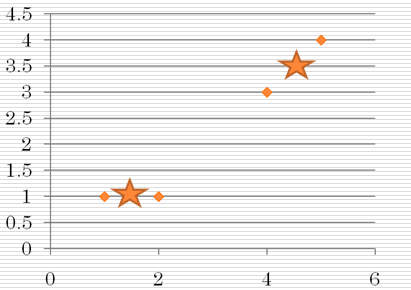
d(B, c1 ) = 1 < d(B, c2 ) = 5.56 => Phân B thuộc cụm 1

d(C, c1 ) = 13 > d(C, c2 ) = 0.22 => Phân C thuộc cụm 2

d(D, c1 ) = 25 > d(D, c2 ) = 3.56 => Phân D thuộc cụm 2

**- Bước 4-2:** Lặp lại bước 3-Cập nhật trọng tâm

c1 = (3/2, 1) và c2 = (9/2, 7/2)



*Hình 2.24. Tính toán lại trọng tâm các cluster*

Do trọng tâm có sự thay đổi, cho nên ta lặp lại bước 2.

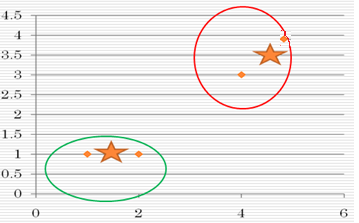
**- Bước 4-3:** Lặp lại bước 2

d(A, c1 ) = 0.25 < d(A, c2 ) = 18.5 => Phân A thuộc cụm 1

d(B, c1 ) = 0.25 < d(B, c2 ) = 12.5 => Phân B thuộc cụm 1

d(C, c1 ) = 10.25 > d(C, c2 ) = 0.5 => Phân C thuộc cụm 2

d(D, c1 ) = 21.25 > d(D, c2 ) = 0.5 => Phân D thuộc cụm 2



*Hình 2.25. Các cluster sau phân hoạch*

Do trọng tâm không có sự thay đổi, cho nên ta dừng lặp. Kết quả phân được 2 cụm, với cụm 1{A,B}, cụm 2{C,D}.

Bài tập 1. Dùng giải thuật k-means để phân 7 điểm ảnh sau thành K cụm (K=2)

X1(4,1) ; X2(5,1) ; X3(5,2) ; X4(1,4) ; X5(1,5) ; X6(2,4) ; X7(2,5)

Bài tập 2. dùng giải thuật k-means để phân 7 điểm ảnh sau thành K cụm (K=3)

X1(4,1) ; X2(5,1) ; X3(5,2) ; X4(1,4) ; X5(1,5) ; X6(2,4) ; X7(2,5)

### 7. Phân vùng bằng phương pháp chia và hợp

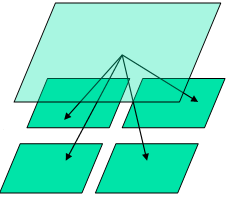
Một phương pháp cũng thường được dùng để phân đoạn ảnh đó là phương pháp Split-and-Merge (Chia và hợp). Giải thuật này như sau:

- Bước 1: Chia

Chia đệ quy các vùng không đồng nhất theo một tiêu chuẩn nào đó (phương sai, khoảng cách lớn nhất - max, nhỏ nhất - min) thành các vùng nhỏ hơn. Chia một vùng thành 4 vùng con, tính chất của vùng con được tính toán theo 1 tiêu chuẩn nào đó.

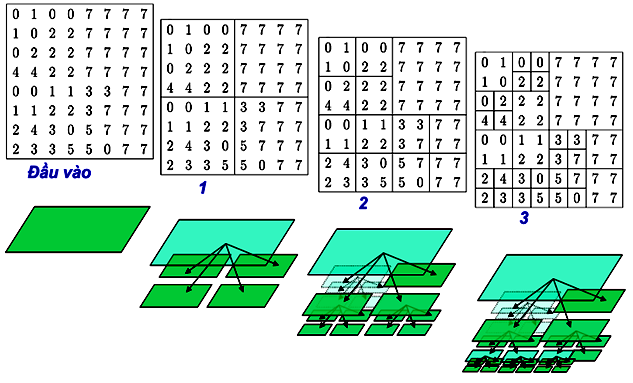
Ảnh có thể biểu diễn thành 1 cây với nút gốc là toàn bộ ảnh (ảnh ban đầu). Mỗi nút F không đồng nhất (Tức là các điểm ảnh trong node có sự sai khác theo phương sai hoặc max/min > epsilon nào đó) được chia thành 4 phần – 4 phần này sẽ trở thành nút con của F.

Giải thuật tiếp tục cho đến khi không còn vùng không đồng nhất nào.



*Hình 2.26. Minh họa phép chia (split) ảnh*

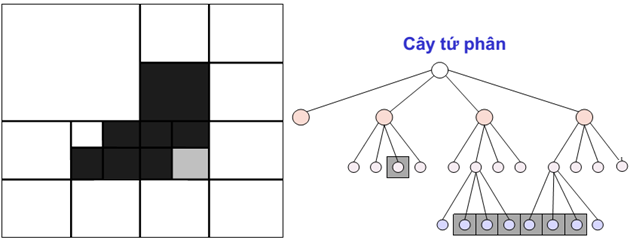
Ví dụ chia ảnh sau với đồng nhất trong vùng được tính theo chênh lệch max-min trong vùng là 1.



*Hình 2.27. Minh họa các bước chia ảnh với chênh lệch max-min là 1*

- Bước 2: Hợp

Gộp các vùng đồng nhất lân cận nhau theo một tiêu chuẩn nào đó.



*Hình 2.28. Minh họa ảnh sau khi hợp theo mô hình cây tứ phân*

Ví dụ đối với bức ảnh đã được chia ở trên với đồng nhất trong vùng được tính theo chênh lệch max-min trong ảnh là 1. Ta gộp ảnh đã chia trên với gộp các vùng lân cận theo chênh lệch max-min trong các vùng liền kề là 1. Kết quả sau khi gộp là:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 7 | 7 | 7 | 7 |
| 1 | 0 | 2 | 2 | 7 | 7 | 7 | 7 |
| 0 | 2 | 2 | 2 | 7 | 7 | 7 | 7 |
| 4 | 4 | 2 | 2 | 7 | 7 | 7 | 7 |
| 0 | 0 | 1 | 1 | 3 | 3 | 7 | 7 |
| 1 | 1 | 2 | 2 | 3 | 7 | 7 | 7 |
| 2 | 4 | 3 | 0 | 5 | 7 | 7 | 7 |
| 2 | 3 | 3 | 5 | 5 | 0 | 7 | 7 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 | 7 | 7 | 7 | 7 |
| 1 | 0 | 2 | 2 | 7 | 7 | 7 | 7 |
| 0 | 2 | 2 | 2 | 7 | 7 | 7 | 7 |
| 4 | 4 | 2 | 2 | 7 | 7 | 7 | 7 |
| 0 | 0 | 1 | 1 | 3 | 3 | 7 | 7 |
| 1 | 1 | 2 | 2 | 3 | 7 | 7 | 7 |
| 2 | 4 | 3 | 0 | 5 | 7 | 7 | 7 |
| 2 | 3 | 3 | 5 | 5 | 0 | 7 | 7 |

*Hình 2.29.**Kết quả sau khi hợp ảnh với chênh lệch max-min là 1*

***Bài tập áp dụng:***

1. Cho ảnh sau, hãy phân vùng ảnh theo phương pháp Split and Merge (Chia và ghép) với phương sai ngưỡng max-min là 2:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 4 | 3 | 6 | 6 | 6 | 6 |
| 1 | 1 | 3 | 3 | 7 | 7 | 7 | 7 |
| 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 |
| 6 | 5 | 6 | 6 | 6 | 6 | 6 | 6 |
| 1 | 2 | 1 | 1 | 4 | 3 | 3 | 3 |
| 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 |
| 5 | 6 | 1 | 1 | 3 | 4 | 1 | 1 |
| 6 | 5 | 2 | 1 | 3 | 3 | 1 | 2 |

1. Cho ảnh sau, hãy phân vùng ảnh theo phương pháp Split and Merge (Chia và ghép) với phương sai ngưỡng max-min là 1:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 4 | 3 | 6 | 6 | 6 | 6 |
| 1 | 1 | 3 | 3 | 7 | 7 | 7 | 7 |
| 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 |
| 6 | 5 | 6 | 6 | 6 | 6 | 6 | 6 |
| 1 | 2 | 1 | 1 | 4 | 3 | 3 | 3 |
| 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 |
| 5 | 6 | 1 | 1 | 3 | 4 | 1 | 1 |
| 6 | 5 | 2 | 1 | 3 | 3 | 1 | 2 |

* + 1. **Dữ liệu video**

Cho 1 CSDL video, yêu cầu truy nhập vào dữ liệu video, xác định và tìm kiếm đoạn video với các yêu cầu người dùng truy vấn như sau:

- Tìm tất cả các cảnh có Denis Dopeman và Jane Shady gặp nhau.

- Tìm tất cả các cảnh có Denis Dopeman cạnh máy bay, ở 1 sân bay trên sa mạc.

Chúng ta cần tổ chức nội dung video và cơ sở dữ liệu video thế nào để có thể truy vấn được?

### Tổ chức nội dung cho 1 video

Để tổ chức được nội dung 1 video và truy vấn được, trước tiên chúng ta cần phải trả lời được các câu hỏi sau:

– Các khía cạnh nào cần được quan tâm khi truy nhập video ? (WHICH)

– Làm thế nào để lưu trữ các đối tượng 1 cách hiệu quả ?(HOW)

– Ngôn ngữ truy vấn ? (QUERY LANGUAGES)

– Làm thế nào để trích chọn được nội dung một cách tự động ? (CONTENT EXTRACTION).

Ví dụ xét 1 video quay 1 giờ học của sinh viên, các điểm mà chúng ta cần phải quan tâm tới đó là:

– Con người: giáo viên, sinh viên

– Hoạt động:

* + Thuyết giảng: GV trình bày 1 vấn đề gì đó.
  + SV đặt câu hỏi/ trả lời câu hỏi, …

- Như vậy các điểm chính với 1 video mà chúng ta cần quan tâm đó là:

* Con người: các nhân vật
* Đồ vật: nhà kho, sân bay, ô tô, …/ Động vật: ngựa, chim, ...
* Hoạt động: cất cánh, hạ cánh, hát, nhảy, …

- Với 1 khung hình (frame) nào đó có:

* + - * tập các đối tượng và hoạt động liên quan
      * Mỗi đối tượng/hoạt động có nhưng thuộc tính nhất định
* Con người: tên, tuổi
* Thuyết giảng: AI trình bày về CÁI GÌ ?

Mục đích của chúng ta là tìm cách biểu diễn các đối tượng và mối liên hệ giữa chúng để việc truy vấn hiệu quả nhất.

### Biểu diễn đối tượng trong 1 video

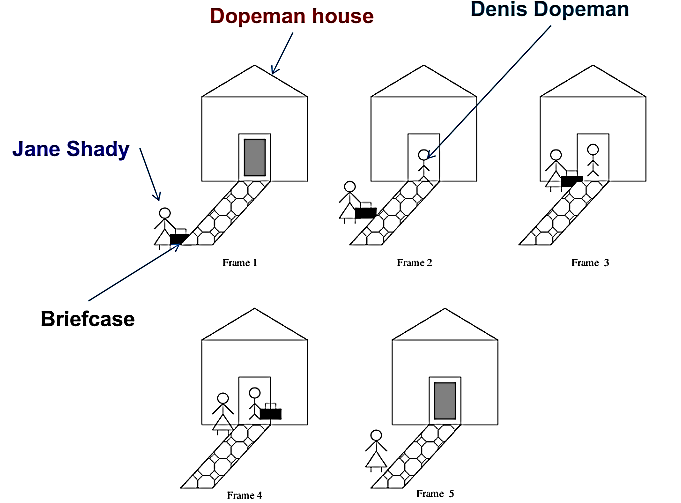
Mỗi đối tượng (Object) được đặc trưng bởi các thuộc tính của đối tượng, mỗi thuộc tính bao gồm tên thuộc tính (pname) và các giá trị của thuộc tính tương ứng (values).

Mỗi đối tượng bao gồm các thuộc tính, các thuộc tính được chia làm 2 loại đó là:

- fd: tập các thuộc tính biến đổi theo mỗi khung hình (framedependent).

- fi: tập các thuộc tính không phụ thuộc vào khung hình (frame-independent).

Ví dụ xét kịch bản tập hợp các Frame video sau:



*Hình 2.36. Kịch bản tập hợp các Frame video*

Thuộc tính frame-dependent:

|  |  |  |
| --- | --- | --- |
| **Frame** | **Objects** | **Frame-dependent Properties** |
| 1 | Jane Shady | has(briefcase), at(path-pront) |
| Dopeman-House | door(closed) |
| Briefcase |  |
| 2 | Jane Shady | has(briefcase), at(path-middle) |
| Denis Dopeman | at(door) |
| Dopeman-House | door(open) |
| Briefcase |  |
| 3 | Jane Shady | has(briefcase), at(door) |
| Denis Dopeman | at(door) |
| Dopeman-House | door(open) |
| Briefcase |  |
| 4 | Jane Shady | at(door) |
| Denis Dopeman | has(briefcase), at(door) |
| Dopeman-House | door(open) |
| Briefcase |  |
| 5 | Jane Shady | at(middle) |
| Dopeman-House | door(closed) |
| Briefcase |  |

*Bảng 2.7. Thuộc tính frame-dependent*

Thuộc tính frame-independent

|  |  |  |
| --- | --- | --- |
| Object | Frame-Independent property | Value |
| Jane Shady | age | 35 |
| height | 170 (cm) |
| dopeman-house | address | 6717 Pimmit Drive Falls Church, VA 22047. |
| type | brick |
| color | brown |
| Denis Dopeman | age | 56 |
| height | 186 |
| briefcase | color | black |
| length | 40 (cm) |
| width | 31 (cm) |

*Bảng 2.8. Thuộc tính frame-independent*

### B.1. Biểu diễn hành động trong 1 video

Một video ngoài có các đối tượng và các thuộc tính của đối tượng, video đó còn có các hành động cần phải được biểu diễn. Ví dụ đối với video ở kịch bản trên, ta có lược đồ hành động (Activity schema - ACT\_SCH) của video như sau:

ExchangeObject = {(Giver, Person), (Receiver, Person),(Item, Thing)}

– Person: tập tất cả nhân vật trong cảnh - Jane Shady, Denis Dopeman

– Thing: tập tất cả các vật có thể trao đổi: briefcase

Ở đây: hành động: Giver = Jane Shady, Receiver = Denis Dopeman, Item = Briefcase.

### B.2. Biểu diễn video

Sau khi các đối tượng và các hành động của 1 video được liệt kê biểu diễn. Bước tiếp theo chúng ta biểu diễn video trong 1 CSDL video. Xét video V có:

– framenum(v): số khung hình trong video

– content(v) = {OBJ, ACT, λ}: nội dung video.

**Ở đây:**

- OBJ = {oid1, oid2,… , oidn}: tập các đối tượng chính được liệt kê trong video.

- ACT = {AcID1, AcID2,… , AcIDk}: tập các sự kiện, hành động chính được liệt kê trong video.

- λ: {1, 2, …, framenum(v)} số các chuyển tiếp nội dung video ~ 2OBJ U ACT

Sau khi video được biểu diễn như trên, chúng ta lại tiếp tục xây dựng một thư viện các video đơn giản:

VidLib = (Vd\_Id, VidContent, framenum, plm, R), ở đây:

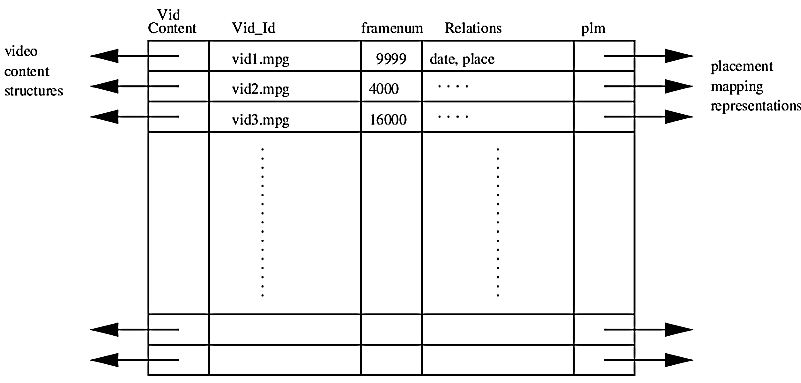
– Vd\_id: tên của video.

– VidContent: nội dung của video.

– framenum: số khung hình trong video.

– Plm (placement maping): chỉ rõ các địa chỉ các phần khác nhau của video.

– R: tập các quan hệ về video để lưu các DL có cấu trúc: date, author, …



*Hình 2.37. Mô hình biểu diễn CSDL Video*

### 8.Truy vấn dữ liệu video

Đối với các video trong CSDL video chúng ta cần phải truy vấn dữ liệu với các kiểu yêu cầu truy vấn như sau:

– Tìm kiếm các đoạn video trong CSDL thỏa mãn điều kiện nào đó.

– Tìm kiếm các đối tượng: cho video v & đoạn [s, e] (startframe, endframe) của v: các đối tượng xuất hiện trong tất cả các khung hình hoặc 1 số khung hình nằm trong [s, e]

– Tìm kiếm các hành động: cho 1 video + đoạn (s, e) của v, tìm các sự kiện xảy ra trong tất cả các khung hình hoặc 1 số khung hình nằm trong [s, e].

– Truy vấn dựa trên thuộc tính: tìm kiếm các video hoặc các đoạn video trong đó các đối tượng/sự kiện thỏa mãn điều kiện nào đó.

Các hàm truy vấn video thường sử dụng:

|  |  |
| --- | --- |
| FindVideoWithObject(o) | {(v,s,e)}:đoạn trong video v có chứa object o ở tất cả các frame thuộc đoạn [s,e]. |
| FindVideoWithActivity(a) | {(v,s,e)}:đoạn trong video v có chứa sự kiện a xảy ra. |
| FindVideoWithActivityandProp(a,p,z) | {(v,s,e)}:đoạn trong video v có chứa sự kiện a với thuộc tính p=z xảy ra. |
| FindVideoWithObjectandProp(o,p,z) | {(v,s,e)}:đoạn trong video v có chứa object o với z là giá trị của thuộc tính p. |
| FindObjectsInVideo(v,s,e) | Các đối tượng xuất hiện trong đoạn [s,e] ϵ v |
| FindActivitiesAndPropInVideo(v,s,e) | Các sự kiện và giá trị thuộc tính tương ứng trong đoạn [s,e] ϵ v |
| FindActivitiesInVideo(v,s,e) | Các sự kiện xảy ra trong đoạn [s,e] ϵ v |
| FindObjectsAndPropInVideo(v,s,e) | Các đối tượng và giá trị thuộc tính tương ứng trong đoạn [s,e] ϵ v |

*Bảng 2.9. Các hàm truy vấn video thường dùng*

**Ngôn ngữ truy vấn thường sử dụng ngôn ngữ SQL mở rộng:**

SELECT Vid:[s,e]

FROM Video <source>

WHERE term IN func\_call

Ví dụ 1: Hãy đưa ra video trong thư viện video VidLib1 mà có nhân vật Denis Dopeman.

SELECT Vid:[s,e]

FROM Video: VidLib1

WHERE (Vid, s, e) IN FindVideoWithObject(Denis Dopeman)

Ví dụ 2: Hãy đưa ra video trong thư viện video VidLib mà có nhân vật Denis Dopeman từ frame 100 đến frame 500.

SELECT Vid:[100,500]

FROM Video: VidLib

WHERE (Vid, 100, 500) IN FindVideoWithObject(Denis Dopeman).

Ví dụ 3: Tìm tất cả đoạn video trong CSDL video trong đó Jane Shady trao 1 chiếc vali cho Denis Dopeman

SELECT Vid:[s,e]

FROM Video: VidLib

WHERE (Vid, s, e) IN FindVideoWithObject(Denis Dopeman)

AND (Vid, s, e) IN FindVideoWithObject(Jane Shady)

AND (Vid, s, e) IN FindVideoWithActivityandProp(ExchangeObject,Giver, Jane Shady)

AND (Vid, s, e) IN FindVideoWithActivityandProp(ExchangeObject,Receiver, Denis Dopeman)

AND (Vid, s, e) IN FindVideoWithActivityandProp(ExchangeObject,Item, Briefcase).

Ví dụ 4. Tìm tất cả đoạn video trong CSDL video trong đó Jane Shady đội mũ màu đỏ.

SELECT Vid:[s,e]

FROM Video: VidLib

WHERE (Vid, s, e) IN FindVideoWithObject(Jane Shady)

AND (Vid, s, e) IN FindVideoWithObjectandProp(Hat,color,red)

AND (Vid,s,e) IN FindVideoWithActivityandProp(Đội mũ, người đội, Jane Shady)

AND (Vid, s, e) IN FindVideoWithActivityandProp(Đội mũ, Được đội, Hat).

# Chương 3: Nén dữ liệu đa phương tiện

## 5.2. Các kỹ thuật nén dữ liệu

### 5.2.1. Nén văn bản

Lưu trữ văn bản không đòi hỏi nhiều dung lượng bộ nhớ như lưu trữ các loại phương tiện khác như image, audio và video nhưng khi cần lưu trữ một số lượng lớn file văn bản thì việc nén dữ liệu văn bản vẫn cần thiết

Đặc trưng chính của nén dữ liệu văn bản là không bị mất thông tin, có nghĩa là có thể phục hồi lại văn bản chính xác như ban đầu sau khi giải nén

Ở đây chúng ta nghiên cứu 3 phương pháp nén dữ liệu:

* + Mã hóa Huffman ( Huffman coding)
  + Mã hóa độ dài hàng loạt ( Run length coding)
  + Mã hóa LZW ( Lempel-Ziv-Welch coding)

### 9. Mã hóa Huffman

Các tập tin của máy tính được lưu dưới dạng các kí tự có chiều dài không đổi là 8 bits (Mã ASCII). Trong nhiều tập tin, xác suất xuất hiện các kí tự này là nhiều hơn các kí tự khác. Nếu chỉ dùng một vài bit để biểu diễn cho các kí tự có xác suất xuất hiện lớn và dùng nhiều bit hơn để biểu diễn cho các kí tự có xác suất xuất hiện nhỏ thì có thể tiết kiệm được độ dài tập tin một cách đáng kể.

Ví dụ: Một file có tổng cộng 1000 kí tự với 4 kí tự khác nhau e, t, x, z. Với xác suất xuất hiện 4 kí tự trong văn bản là e: 0.8, t: 0.16, x: 0.02, z: 0.02.

* + So với cách mã hóa thông thường, để biểu diễn mỗi kí tự cần 2 bít, như vậy 1000 kí tự trên sẽ cần 2000 bít để mã hóa.
  + Với mã hóa Huffman ta gán các chuỗi bit ngắn nhất cho các kí tự được dùng phổ biến nhất, ta gán: e là 1, t là 01, x là 001, z là 000. Như vậy Số bít cần thiết để biểu diễn file là: 1000\*(1\*0.8+2\*0.16+3\*0.02+3\*0.02)= 1240 bít.

Quy tắc để để gán các bít (codes) cho các kí tự được gọi là Bảng mã hóa (codebook).

Ưu điểm của mã hóa Huffman: hệ số nén tương đối cao, phương pháp thực hiện tương đối đơn giản, đòi hỏi ít bộ nhớ, có thể xây dựng dựa trên các mảng bé hơn 64KB. Nhược điểm của phương pháp này là chứa cả bảng mã vào tập tin nén thì phía nhận mới có thể giải mã được do đó hiệu suất nén chỉ cao khi ta thực hiện nén các tập tin lớn.

**Thuật toán nén:**

Input:chuỗi ký tự cần nén

Output: Chuỗi mã hóa chuỗi ký tự nén và bảng mã

**Method**

Bước 1: Tìm hai ký tự có trọng số nhỏ nhất ghép lại thành một, trọng số của ký tự mới bằng tổng trọng số của hai ký tự đem ghép.

Bước 2: Trong khi số lượng ký tự trong danh sách còn lớn hơn một thì thực hiện bước một, nếu không thì thực hiện bước ba.

Bước 3: Tách ký tự cuối cùng và tạo cây nhị phân với quy ước bên trái mã 0, bên phải mã 1.

**Thuật toán giải nén:**

Input: Chuỗi mã hóa chuỗi ký tự nén và bảng mã

Output: Chuỗi gốc ban đầu được giải nén

**Method**

Bước 1: Đọc lần lượt từng bit trong tập tin nén và duyệt cây nhị phân đã được xác định cho đến khi hết một lá. Lấy ký tự ở lá đó ghi ra tệp giải nén.

Bước 2: Trong khi chưa hết tập tin nén thì thực hiện bước một, ngược lại thì thực hiện bước 3.

Bước 3: Kết thúc thuật toán.

Ví dụ: Cho bảng tần suất của 5 chữ cái A,B,C,D,E tương ứng là:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | E |
| 0.10 | 0.15 | 0.30 | 0.16 | 0.29 |

Quá trình xây dựng cây Huffman như sau:

0.30

0.16

0.29

0.29

0.16

0.30

0.15

0.10

Bước 1

0.10

0.15

Bước 2

0.16

0.10

0.15

0.16

0.30

0.29

0.30

0.29

0.10

0.15

Bước 4

Bước 3

Như vậy bộ mã tối ưu tương ứng là

0.29

0.30

0.16

0.10

0.15

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***A*** | ***B*** | ***C*** | ***D*** | ***E*** |
| ***010*** | ***011*** | ***11*** | ***00*** | ***10*** |

Bước 5

***Bài tập:*** Nén chuỗi văn bản sau bằng mã Huffman “xin chao ha noi“

### 