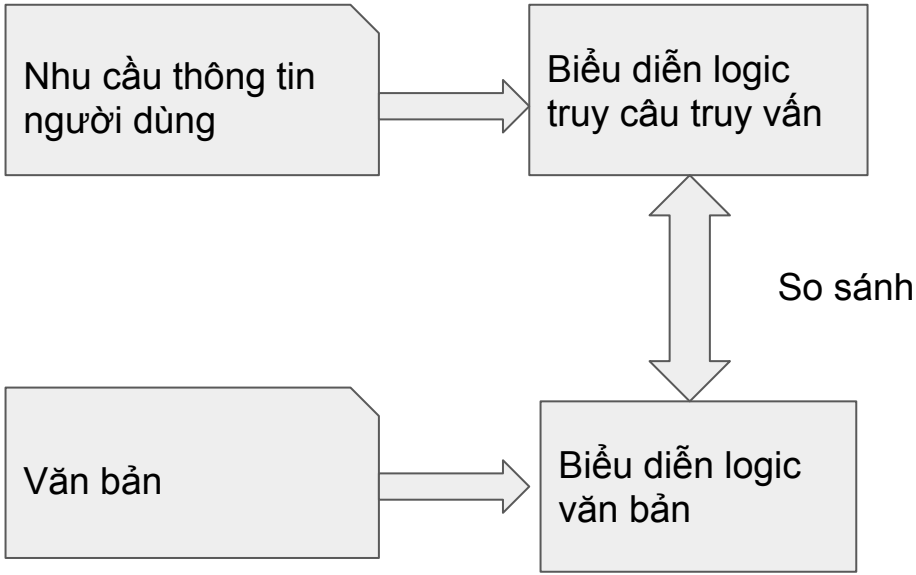
# Tìm kiếm và trình diễn thông tin

Probabilistic information retrieval

# Nội dung chính

1. Ứng dụng lý thuyết xác suất trong tìm kiếm
2. Mô hình nhị phân độc lập
3. Mô hình (Okapi) BM25

# Lý thuyết xác suất trong tìm kiếm thông tin




Không bảo toàn ngữ nghĩa

Văn bản trả về không chắc chắn là văn bản phù hợp

Có thể ứng dụng lý thuyết xác suất trong tìm kiếm thông tin.

# Tổng quan các mô hình xác suất

- Các mô hình xác suất cổ điển:
  - Nguyên tắc xếp hạng xác suất
    - Mô hình nhị phân độc lập,
    - BestMatch25(Okapi)
    - …
- Tìm kiếm văn bản sử dụng mạng Bayes;
- Các mô hình ngôn ngữ
  - Hướng nghiên cứu mới, hiệu năng cao;

Phương pháp xác suất là một trong những phương pháp đã tồn tại từ lâu nhưng vẫn là đề tài nóng trong tìm kiếm thông tin hiện đại.

# Xếp hạng xác suất

Ký hiệu $R_{d,q}$: một biến nhị phân ngẫu nhiên:

$R_{d,q}$ = 1 nếu d phù hợp với q;

$R_{d,q}$ = 0, nếu ngược lại.

Theo phương pháp xếp hạng xác suất, các văn bản được trả về theo thứ tự giảm dần giá trị xác suất văn bản phù hợp với truy vấn: P(R=1|d, q).

# Trọng số từ

Xếp hạng xác suất: Probabilistic Ranking

“Trọng số của từ xuất hiện trong những văn bản đã biết là phù hợp có giá trị cao hơn trọng số của từ đó trong trường hợp không biết những văn bản phù hợp này.”

“Có thể xây dựng cách tính trọng số từ dựa trên giả thuyết về phân bố từ vựng và luật Bayes.”

# Lý thuyết xác suất căn bản (1)

- For events *A* and *B:*

$$p(A,B) = p(A \cap B) = p(A \mid B)p(B) = p(B \mid A)p(A)$$

- Bayes' Rule

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)} = \frac{p(B \mid A)p(A)}{\sum_{X=A,\bar{A}} p(B \mid X)p(X)}$$

Prior

Posterior

- Odds:

$$O(A) = \frac{p(A)}{p(\bar{A})} = \frac{p(A)}{1 - p(A)}$$

# Mô hình nhị phân độc lập

Nhị phân: Văn bản được biểu diễn như vector nhị phân đánh dấu sự xuất hiện của từ

- d = (x1, ... , xn )
- xi =1 nếu thuật ngữ thứ i xuất hiện trong d, 0 nếu ngược lại

Độc lập: Sự xuất hiện của mỗi từ trong văn bản là độc lập với những từ còn lại;

Những văn bản khác nhau có thể có cùng một biểu diễn vector.

# Mô hình nhị phân độc lập (1)

- Cho truy vấn q
  - Với mỗi văn bản d cần tính xác suất d là tài liệu phù hợp p(R=1|q, d)
  - Chỉ quan tâm tới thứ hạng
- Sử dụng cơ hội (Odds) và luật Bayes

- Bayes' Rule

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)} = \frac{p(B \mid A)p(A)}{\sum_{X=A,\bar{A}} p(B \mid X)p(X)}$$

Prior

Posterior

$$O(A) = \frac{p(A)}{p(\bar{A})} = \frac{p(A)}{1-p(A)}$$

$$O(R|q,d) = \frac{p(R=1|q,d)}{p(R=0|q,d)} = \frac{\frac{p(R=1|q)p(d|R=1,q)}{p(d|q)}}{\frac{p(R=0|q)p(d|R=0,q)}{p(d|q)}}$$

# Mô hình nhị phân độc lập (1)

- Cho truy vấn q
  - Với mỗi văn bản d cần tính xác suất d là tài liệu phù hợp p(R=1|q, d)
  - Chỉ quan tâm tới thứ hạng
- Sử dụng cơ hội (Odds) và luật Bayes

$$O(A) = \frac{p(A)}{p(\overline{A})} = \frac{p(A)}{1 - p(A)}$$

$$O(R|q,d) = \frac{p(R=1|q,d)}{p(R=0|q,d)} = \frac{\frac{p(R=1|q)\,p(d|R=1,q)}{p(d|q)}}{\frac{p(R=0|q)\,p(d|R=0,q)}{p(d|q)}}$$

$$O(R|q,d) = \frac{p(R=1|q,d)}{p(R=0|q,d)} = \frac{p(R=1|q)}{p(R=0|q)} \cdot \frac{p(d|R=1,q)}{p(d|R=0,q)}$$

**Hằng số với một truy vấn**

**Cần xác định**

# Mô hình nhị phân độc lập (2)

$$O(R|q,d) = \frac{p(R=1|q,d)}{p(R=0|q,d)} = \frac{p(R=1|q)}{p(R=0|q)} \cdot \frac{p(d|R=1,q)}{p(d|R=0,q)}$$

**Hằng số với một truy vấn**

**Cần xác định**

Sử dụng giả thuyết độc lập

$$\frac{p(d|R=1,q)}{p(d|R=0,q)} = \prod_{i=1}^{n} \frac{p(x_i|R=1,q)}{p(x_i|R=0,q)}$$

$$O(R|q,d) = O(R|q) \cdot \prod_{i=1}^{n} \frac{p(x_i|R=1,q)}{p(x_i|R=0,q)}$$

# Mô hình nhị phân độc lập (3)

$$p_i = p(x_i = 1 | R = 1, q)$$
$$1 - p_i = p(x_i = 0 | R = 1, q)$$
$$r_i = p(x_i = 1 | R = 0, q)$$
$$1 - r_i = p(x_i = 0 | R = 0, q)$$

$$O(R|q,d) = O(R|q) \cdot \prod_{i=1}^{n} \frac{p(x_i|R=1,q)}{p(x_i|R=0,q)}$$

$$= O(R|q) \cdot \prod_{x_i=1} \frac{p(x_i=1|R=1,q)}{p(x_i=1|R=0,q)} \prod_{x_i=0} \frac{p(x_i=0|R=1,q)}{p(x_i=0|R=0,q)}$$

$p_i$: the probability of a term appearing in a document relevant to the query
$r_i$: the probability of a term appearing in an irrelevant document to the query

$$O(R|q,d) = O(R|q) \cdot \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} \cdot \prod_{q_i=1} \frac{1-p_i}{1-r_i}$$

**Từ truy vấn có trong văn bản**

**Tất cả từ truy vấn**

# Mô hình nhị phân độc lập (3)

$$p_i = p(x_i = 1 | R = 1, q)$$
$$1 - p_i = p(x_i = 0 | R = 1, q)$$
$$r_i = p(x_i = 1 | R = 0, q)$$
$$1 - r_i = p(x_i = 0 | R = 0, q)$$

$$O(R|q,d) = O(R|q) \cdot \prod_{i=1}^{n} \frac{p(x_i|R=1,q)}{p(x_i|R=0,q)}$$

$$= O(R|q) \cdot \prod_{x_i=1} \frac{p(x_i=1|R=1,q)}{p(x_i=1|R=0,q)} \prod_{x_i=0} \frac{p(x_i=0|R=1,q)}{p(x_i=0|R=0,q)}$$

$p_i$: the probability of a term appearing in a document relevant to the query
$r_i$: the probability of a term appearing in an irrelevant document to the query

$$O(R|q,d) = O(R|q) \cdot \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} \cdot \prod_{q_i=1} \frac{1-p_i}{1-r_i}$$

**Từ truy vấn có trong văn bản**

**Tất cả từ truy vấn**

# Mô hình nhị phân độc lập (6)

$$p_i = p(x_i = 1 | R = 1, q)$$
$$1 - p_i = p(x_i = 0 | R = 1, q)$$
$$r_i = p(x_i = 1 | R = 0, q)$$
$$1 - r_i = p(x_i = 0 | R = 0, q)$$

Kết quả tìm kiếm được xác định dựa trên

$$RSV(d,q) = \log \prod_{x_i = q_i = 1} \frac{p_i(1-r_i)}{r_i(1-p_i)} = \sum_{x_i = q_i = 1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

$$RSV(d,q) = \sum_{x_i = q_i = 1} c_i; \quad c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

ci có vai trò như trọng số thuật ngữ trong mô hình này

# Những số liệu thống kê cơ bản

s: số lượng văn bản chứa xi, phù hợp với câu query
n: số lượng văn bản chứ xi
S: số lượng văn bản phù hợp
N: số lượng văn bản trong kho dữ liệu

Đại lượng thống kê ứng với từ thứ i:

| Từ/văn bản | Phù hợp | Không phù hợp | Tổng |
|---|---|---|---|
| $x_i=1$ | $s$ | $n-s$ | $n$ |
| $x_i=0$ | $S-s$ | $N-n-S+s$ | $N-n$ |
| Tổng | $S$ | $N-S$ | $N$ |

$$p_i = p(x_i=1|R=1,q)$$
$$1-p_i = p(x_i=0|R=1,q)$$
$$r_i = p(x_i=1|R=0,q)$$
$$1-r_i = p(x_i=0|R=0,q)$$

$$p_i \approx \frac{s}{S} \quad r_i \approx \frac{n-s}{N-S} \quad p_i = p(x_i=1|R=1,q);\ r_i = p(x_i=1|R=0,q);$$

$$c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)} \quad c_i \approx K(N,n,S,s) = \log \frac{s/(S-s)}{(n-s)/(N-n-S+s)}$$

# Làm mịn trọng số

Có thể thêm 0.5 vào mỗi tham số để đảm bảo các trọng số không trở thành vô cùng khi S, s nhỏ:

$$c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)} \quad c_i \approx K(N,n,S,s) = \log \frac{s/(S-s)}{(n-s)/(N-n-S+s)}$$

$$c_t = \log \frac{(s+0.5)(N-S-n+s+0.5)}{(n-s+0.5)(S-s+0.5)}$$

# Bắt đầu thực hiện truy vấn

Hoàn toàn không biết về R, số lượng văn bản phù hợp chiếm tỷ lệ rất nhỏ trong tập tài liệu

$$c_t = \log \frac{(s+0.5)(N-S-n+s+0.5)}{(n-s+0.5)(S-s+0.5)}$$

$$c_t = \log \frac{N-n+0.5}{n+0.5}$$

Trong trường hợp này $c_t$ tương tự trọng số idf.

Có thể sử dụng giá trị này để tính hạng ban đầu

# Bắt đầu thực hiện truy vấn

**d** Biểu diễn vec-tơ văn bản

| | a | b | c | d | e | f | g | h | k | l |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | | | 1 | | | | 1 | 1 | |
| 2 | | | | | | | | 1 | 1 | 1 |
| 3 | | 1 | | | | 1 | 1 | | | |
| 4 | 1 | | | 1 | | | | | | 1 |
| 5 | | | | | | | | 1 | 1 | |
| 6 | | | 1 | | 1 | | | | | |

$$c_t = \log \frac{N - n + 0.5}{n + 0.5}$$

# Cải thiện xếp hạng bằng cách ước lượng pi

- Phản hồi từ người dùng
- Sử dụng hằng số pi = 0.5

# Cải thiện xếp hạng bằng cách ước lượng pi

Phù hợp phản hồi giả lập

1. Giả sử pi là hằng số với mọi xi trong truy vấn.
Ví dụ, pi = 0.5 với văn bản bất kỳ, tính ri và RSV(d,q).
Lưu ý, ci lúc này tương tự idf.

2. Giả sử tập văn bản phù hợp V là tập chứa những văn bản được xếp hạng cao nhất theo mô hình này (dựa vào giá trị RSV).

3. Sử dụng phân bố từ trong V, xác định lại pi và ri

Đặt Vi là tập văn bản có chứa xi, chúng ta có

$$pi = (|Vi| + 0.5) / (|V| + 1)$$

Giả sử không được trả về đồng nghĩa với không phù hợp,

$$ri = (ni – |Vi| + 0.5) / (N – |V| + 1)$$

5 Lặp các bước 2-4 cho tới khi hội tụ và trả về kết quả

# Bài tập

$$c_t = \log \frac{N - n + 0.5}{n + 0.5}$$

$$RSV(d,q) = \sum_{x_i = q_i = 1} c_i;$$

Cho các văn bản sau:

**Doc1: [breakthrough for schizophrenia]**

**Doc2: [new schizophrenia drug]**

**Doc3: [new approach for treatment of schizophrenia]**

**Doc4: [new hopes for schizophrenia patients]**

b) Các văn bản nào sẽ được trả về cho truy vấn:

**schizophrenia drug**

# Ví dụ trọng số phù hợp

**d** Biểu diễn vec-tơ văn bản

| | a | b | c | d | e | f | g | h | k | l |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | | | 1 | | | | 1 | 1 | |
| 2 | | | | | | | | 1 | 1 | 1 |
| 3 | | 1 | | | | 1 | 1 | | | |
| 4 | 1 | | | 1 | | | | | | 1 |
| 5 | | | | | | | | 1 | 1 | |
| 6 | | | 1 | | 1 | | | | | |

$$c_t = \log \frac{(s+0.5)(N-S-n+s+0.5)}{(n-s+0.5)(S-s+0.5)}$$

# Tổng kết mô hình BIM

- Mô hình xác suất dựa trên lý thuyết xác suất để mô hình hóa sự không chắc chắn trong quá trình tìm kiếm
- Sử dụng các giả thuyết về sự độc lập trong quá trình ước lượng giá trị xác suất
- Từ không xuất hiện trong truy vấn không ảnh hưởng tới tính phù hợp (có pi = ri)
- Trọng số ban đầu của thuật ngữ khi chưa có thông tin về văn bản phù hợp được xác định tương tự idf.
- Phù hợp phản hồi giả lập có thể giúp cải thiện xếp hạng bằng cách xác định lại xác suất thuật ngữ
- Không sử dụng các tần suất thuật ngữ  văn bản
- BM25

# “Early” versions of BM25

- Version 1: using the saturation function

$$c_i^{BM25v1}(tf_i) = c_i^{BIM}\frac{tf_i}{k_1 + tf_i}$$

$$RSV(d,q) = \sum_{x_i = q_i = 1} c_i;$$

- Version 2: BIM simplification to IDF

$$c_i^{BM25v2}(tf_i) = \log\frac{N}{df_i} \times \frac{(k_1+1)tf_i}{k_1 + tf_i}$$

  - $(k_1+1)$ factor doesn’t change ranking, but makes term score $1$ when $tf_i = 1$
  - Similar to *tf-idf*, but term scores are bounded

# Document length normalization

- Longer documents are likely to have larger $tf_i$ values
- Why might documents be longer?
  - Verbosity: suggests observed $tf_i$ too high
  - Larger scope: suggests observed $tf_i$ may be right
- A real document collection probably has both effects
- … so should apply some kind of partial normalization

# Document length normalization

- Document length:

$$dl = \sum_{i \in V} tf_i$$

- $avdl$: Average document length over collection
- Length normalization component

$$B = \left( (1-b) + b \frac{dl}{avdl} \right), \qquad 0 \leq b \leq 1$$

  - $b = 1$ full document length normalization
  - $b = 0$ no document length normalization

# Okapi BM25

- Normalize *tf* using document length

$$tf_i' = \frac{tf_i}{B}$$

$$c_i^{BM25}(tf_i) = \log\frac{N}{df_i} \times \frac{(k_1+1)tf_i'}{k_1 + tf_i'}$$

$$= \log\frac{N}{df_i} \times \frac{(k_1+1)tf_i}{k_1((1-b)+b\frac{dl}{avdl}) + tf_i}$$

- BM25 ranking function

$$RSV^{BM25} = \sum_{i \in q} c_i^{BM25}(tf_i);$$

# Okapi BM25

$$RSV^{BM25} = \sum_{i \in q} \log \frac{N}{df_i} \cdot \frac{(k_1 + 1) tf_i}{k_1((1-b) + b \frac{dl}{avdl}) + tf_i}$$

- $k_1$ controls term frequency scaling
  - $k_1 = 0$ is binary model; $k_1$ large is raw term frequency
- $b$ controls document length normalization
  - $b = 0$ is no length normalization; $b = 1$ is relative frequency (fully scale by document length)
- Typically, $k_1$ is set around 1.2–2 and $b$ around 0.75

# Why is BM25 better than VSM tf-idf?

- Suppose your query is [machine learning]
- Suppose you have 2 documents with term counts:
  - doc1: learning 1024; machine 1
  - doc2: learning 16; machine 8
- tf-idf: $\log_2$ tf * $\log_2$ (N/df)
  - doc1: 11 * 7 + 1 * 10 = **87**
  - doc2: 5 * 7 + 4 * 10 = **75**
- BM25: $k_1 = 2$
  - doc1: 7 * 3 + 10 * 1 = **31**
  - doc2: 7 * 2.67 + 10 * 2.4 = **42.7**

# Bài tập

**D1. 'Human machine interface for lab abc computer applications',**

**D2. 'A survey of user opinion of computer system response time',**

**D3. 'The EPS user interface management system',**

**D4. 'System and human system engineering testing of EPS',**

**D5. 'Relation of user perceived response time to error measurement',**

**D6. 'The generation of random binary unordered trees',**

**D7. 'The intersection graph of paths in trees',**

**D8. 'Graph minors IV Widths of trees and well quasi ordering',**

**D9. 'Graph minors A survey'**

**stopwords = ['for', 'a', 'of', 'the', 'and', 'to', 'in']**

**query = 'The intersection of graph survey and trees'**