Classic Information Retrieval

1

Information Retrieval

- User wants information from a collection of "objects": information need
- User formulates need as a "query"
  - Language of information retrieval system
- System finds objects that "satisfy" query
- System presents objects to user in "useful form"
- User determines which objects from among those presented are relevant

➢ Define each of the words in quotes

2

Information Retrieval

- User wants information from a collection of "objects": information need
- User formulates need as a "query"
  - Language of information retrieval system
- System finds objects that "satisfy" query
- System presents objects to user in "useful form"
- User determines which objects from among those presented are relevant CRITICAL NOTION

➢ Define each of the words in quotes
➢ Develop algorithms

3

Think first about text documents

Although search has changed, classic techniques still provide foundations
– our starting point

- Early digital searches – digital card catalog:
  - subject classifications, keywords
- "Full text" : words + natural language syntax
  - No "meta-structure"
- Classic study
  - Gerald Salton SMART project 1960's

4

Scaling

- What are attributes changing from 1960's to online searches of today?
- How do they change problem?

5

Develop models

Begin with document:

How do we view document contents?

6

Modeling: "*query*"

How do we want to express a query?

What does it mean?

7

Modeling: "*query*"

We will consider

- *Query*
  - Basic query is one term
  - Multi-term query is (choose one):
    - Set of terms
    - Sequence of terms
      - multiplicity?
      - Other constraints?
    - Boolean combination of terms

8

Modeling: "*satisfying*"

- What determines if document satisfies query?
- That depends ….
  - Document model
  - Query model
  - definition of "satisfying" can still vary
- *START SIMPLE*
  - *better understanding*
  - *Use components of simple model later*

9

Present results in *"useful form"*

- most basic: give list of results
- meaning of order of list? => RANKING
- Goals of ranking
  - Order documents that satisfy a query by how well match the query
  - Capture relevance to user by algorithmic method of ordering

10

(pure) Boolean Model of IR

- Document: *set* of terms
- Query: Boolean expression over terms
- Satisfying:
  - Doc. evaluates to "true" on single-term query if contains term
  - Evaluate doc. on expression query as you would any Boolean expression
  - doc satisfies query if evals to true on query

11

Boolean Model example

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Query: (principles OR knowledge) AND (science AND NOT(engineering))

12

Boolean Model example

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Query: (principles OR knowledge) AND (science AND NOT(engineering))

Doc 1: (0 OR 1) AND (1 AND NOT(0)) TRUE

13

Boolean Model example

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Query: (principles OR knowledge) AND (science AND NOT(engineering))

Doc 2: (1 OR 0) AND (1 AND NOT(1)) FALSE

14

(pure) Boolean Model of IR: how "*present results in useful form*"

- most basic: give list of results
- meaning of order of list? => RANKING?

- There is no sense of ranking in pure Boolean model
  - need idea in addition to "satisfying documents": generalize model

15

Restrict Boolean Model

- AND model: query is the AND of a set of query terms: term_1 AND term_2 AND…
  - just need specify set of terms
  - This model used by current search engines
- OR model: query is the OR of a set of query terms: term_1 OR term_2 OR …
  - just need specify set of terms
  - This original model for IR development
    - why?

16

Simple Model with Ranking

- Document: *bag* of terms - count occurrences
- Query: *set* of terms
- Satisfying: OR model
- Ranking: numerical score measuring degree to which document satisfies query

  some choices:
    - one point for each query term in document
    - ➢one point for ***each occurrence*** of a query term in document

- Documents returned in sorted list by decreasing score

17

Simple Model: example

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Frequencies:

science 1; knowledge 2; principles 0; engineering 0

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Frequencies:

science 2; knowledge 0; principles 1; engineering 1

18

Generalize Simple Model: The Vector Model

- Have a *lexicon* (aka *dictionary*) of all terms appearing in the collection of documents
  - m terms in all, number 1, …, m
- Document: an m-dimensional vector
  - i^{th} entry of the vector is a real-valued *weight* (importance of) term i in the document
- Query: an m-dimensional vector
  - The i^{th} entry of the vector is a real-valued *weight* (importance of) term i in the query

19

Vector Model: Satisfying & Ranking

- Satisfying:
  - Each document is scored as to the degree it satisfies query (higher better)
  - there is no inherent notion of satisfying
  - typically doc satisfies query if score is > threshold
- Ranking:
  - Documents are returned in sorted list decreasing by score:
    - Include only highest n documents, some n?

20

Where get dictionary of t terms?

- Pre-determined dictionary.
  - How sure get all terms?
- Build lexicon when collect documents
  - What if collection dynamic: add terms?

21

How compute score

Calculate a vector function of the document vector and the query vector

Choices:

1. distance between the vectors:

$$\text{Dist}(\boldsymbol{d},\boldsymbol{q}) = \sqrt{\left(\Sigma^{t}_{i=1}(\boldsymbol{d}_i - \boldsymbol{q}_i)^2\right)}$$

- Is *dissimilarity* measure
- Not normalized: Dist ranges [0, inf.)
- Fix: use $e^{-\textbf{Dist}}$ with range (0,1]
- Is it the right sense of difference?

22

How compute score, continued

2. angle between the vectors:

Dot product: $\boldsymbol{d} \bullet \boldsymbol{q} = \Sigma^{t}_{i=1}(\boldsymbol{d}_i * \boldsymbol{q}_i)$

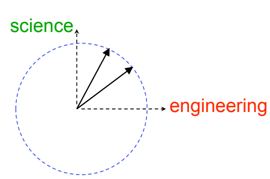
- Is *similarity* measure
- Not normalized: dot product ranges (-inf., inf.)
- Fix: use normalized dot product, range [-1,1]

  $(\boldsymbol{d} \bullet \boldsymbol{q}) / (|\boldsymbol{d}| * |\boldsymbol{q}|)$ where $|\boldsymbol{v}| = \sqrt{\Sigma^{t}_{i=1}(\boldsymbol{v}_i^2)}$ the length of $\boldsymbol{v}$
- In practice vector components are non-negative so range is [0,1]
- This most commonly used function for score

23

Normalizing vectors

- If use unit vectors, $\boldsymbol{d} / |\boldsymbol{d}|$ and $\boldsymbol{v} / |\boldsymbol{v}|$ some of issues go away



24

The Simple Model as a Vector Model

- Document: an *m*-dimensional vector
  - i^{th} entry of the vector is the number of times term *i* appears in the document
- Query: an *m*-dimensional vector
  - The i^{th} entry of the vector is 1 if term *i* in the query, 0 otherwise
- Vector function: dot product

25

How compute weights d_i and q_i?

First:

observations about this model?

26

Vector model: Observations

- Have matrix of terms by documents
  ⇒Can use linear algebra
- Queries and documents are the same
  ⇒Can compare documents same way
  - Clustering documents
- Document with only some of query terms can score higher than document with all query terms

27

How compute weights

- Vector model *could* have weights assigned by human intervention
  - may add meta-information
  - User setting query weights might make sense
    - User decides importance of terms in own search
  - Humans setting document weights?
    - Who? Billions+ of documents
- Return to model of documents as bag of words – calculate weights
  - Function mapping bag of words to vector

28

Calculations on board

29

Summary weight calculation

- General notation:
  - w_{jd} is the weight of term *j* in document *d*
  - $freq_{jd}$ is the # of times term *j* appears in doc *d*
  - n_j = # docs containing term j
  - N = number of docs in collection
- Classic *tf-idf* definition of weight:

  $w_{jd} = freq_{jd} * \log(N/ n_j)$

30

Summary weight calculation

- General notation:
  - w_{jd} is the weight of term *j* in document *d*
  - $freq_{jd}$ is the # of times term *j* appears in doc *d*
  - n_j = # docs containing term j
  - *N* = number of docs in collection
- Classic *tf-idf* definition of weight, normalized:

$$u_{jd} = freq_{jd} \; * \; \log(N/\, n_j)$$

$$w_{jd} = \frac{u_{jd}}{(\Sigma^{t}_{i=1}(u_{id}^2))^{1/2}}$$

31

Weight of query components?

- Set of terms, ***some choices***:
  1. w_{jq} = 0 or 1
  2. $w_{jq} = freq_{jq} \; * \; \log(N/\, n_j)$
     $= 0$ *or* $\log(N/\, n_j)$
- Bag of terms
  - Analyze like document
    Some queries are prose expressions of ***information need***

Do we want idf term in both document weights and query weights?

32

Vector Model example

Doc 1: "Computers have brought the world to our fingertips. We will try to understand at a basic level the science -- old and new -- underlying this new Computational Universe. Our quest takes us on a broad sweep of scientific knowledge and related technologies… Ultimately, this study makes us look anew at ourselves -- our genome; language; music; "knowledge"; and, above all, the mystery of our intelligence. (cos 116 description)

Frequencies:
science 1; knowledge 2; principles 0; engineering 0

Doc 2: "An introduction to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science …" (cos 126 description)

Frequencies:
science 2; knowledge 0; principles 1; engineering 1

33

Vector model example cont.

- Consider the 5 100-level and 200-level COS courses as the collection (109, 217, 226)
- Only other appearance of our 4 words is "science" once in 109 description.
- idf:
  science ln(5/3) = .51
  engineering, principles, knowledge:
  ln(5/1) = 1.6

34

Term by Doc. Table: $freq_{jd} \; * \; \log(N/\, n_j)$

	Doc 1	Doc 2
science	.51	1.02
engineering		1.6
principles		1.6
knowledge	3.2	

35

Unnormalized dot product for query: *science, engineering, knowledge, principles* using 0/1 query vector

- Doc 1: 3.71
- Doc 2: 4.22
- If documents have about same vector length, this right ratio for normalized (cosine) score

36

Additional ways to calculate document weights

- Dampen frequency effect:
  w_{jd} = 1+ log ($freq_{jd}$) if $freq_{jd}$ > 0; 0 otherwise
- *Use smoothing term to dampen effect:*
  $W_{jd} = a + (1\text{-}a)\ freq_{jd} / max_p\ (freq_{pd})$
  - *a is typically .4 or .5*
  - *Can multiply second term by idf*
- *Effects for long documents (Section 6.4.4)*

37

Classic IR models - Taxonomy

Well-specified models:

✓Boolean
✓Vector

- Probabilistic
  – based on probabilistic model of words in documents

38