CIS 833 – Information Retrieval and Text Mining

Lecture 13

Probabilistic Models

October 6, 2015

Credits for slides: Hofmann, Mihalcea, Mobasher, Mooney, Schutze.

Assignments

- PA1 due October 16th (extended)
- Exam 1 – October 13th
- Exam review – October 7th

Classes of Retrieval Models

- Boolean models (set theoretic)
  - Extended Boolean
- Vector space models (algebraic)
  - Generalized VS
  - Latent Semantic Indexing
- Probabilistic models
  - Inference Networks
  - Belief Networks

Exact match

Ranking - "Best" match

Required Reading

Probabilistic Retrieval Models

- Chapter 11: 11.2-11.4 - Probabilistic retrieval models

Binary Independence Model

- Traditionally used in conjunction with PRP
- **"Binary" = Boolean**: documents are represented as binary incidence vectors of terms:
  - $\vec{x} = (x_1, \ldots, x_n)$
  - $x_i = 1$ <u>iff</u> term *i* is present in document *d* having representation *x*.
- **"Independence":** terms occur in documents independently
  - Different documents can be modeled as same vector
- Bernoulli Naive Bayes model (cf. text categorization!)

Binary Independence Model

- Given query ***q***:

$$O(R \mid q, \vec{x}) = \frac{p(R \mid q, \vec{x})}{p(NR \mid q, \vec{x})} = \frac{p(R \mid q)}{p(NR \mid q)} \cdot \frac{p(\vec{x} \mid R, q)}{p(\vec{x} \mid NR, q)}$$

Constant for a given query

Needs estimation

Using **Independence** Assumption:

$$\frac{p(\vec{x} \mid R, q)}{p(\vec{x} \mid NR, q)} = \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$$

So : $$O(R \mid q, \vec{x}) = O(R \mid q) \cdot \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$$

Binary Independence Model

$$O(R \mid q, \vec{x}) = O(R \mid q) \cdot \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$$

- Since x_i is either 0 or 1:

$$O(R \mid q, \vec{x}) = O(R \mid q) \cdot \prod_{x_i = 1} \frac{p(x_i = 1 \mid R, q)}{p(x_i = 1 \mid NR, q)} \cdot \prod_{x_i = 0} \frac{p(x_i = 0 \mid R, q)}{p(x_i = 0 \mid NR, q)}$$

- Let $p_i = p(x_i = 1 \mid R, q); \quad r_i = p(x_i = 1 \mid NR, q);$

- Assume, for all terms not occurring in the query ($q_i=0$) $\quad p_i = r_i$

Then...

Binary Independence Model

$$O(R \mid q, \vec{x}) = \boxed{O(R \mid q)} \cdot \underbrace{\prod_{x_i = q_i = 1} \frac{p_i}{r_i}}_{\text{All matching terms}} \cdot \underbrace{\prod_{\substack{x_i = 0 \\ q_i = 1}} \frac{1 - p_i}{1 - r_i}}_{\text{Non-matching query terms}}$$

$$= \boxed{O(R \mid q)} \cdot \underbrace{\prod_{x_i = q_i = 1} \frac{p_i (1 - r_i)}{r_i (1 - p_i)}}_{\text{All matching terms}} \cdot \underbrace{\prod_{q_i = 1} \frac{1 - p_i}{1 - r_i}}_{\text{All query terms}}$$

Binary Independence Model

$$O(R \mid q, \vec{x}) = \boxed{O(R \mid q)} \cdot \boxed{\prod_{x_i = q_i = 1} \frac{p_i(1-r_i)}{r_i(1-p_i)}} \cdot \boxed{\prod_{q_i = 1} \frac{1-p_i}{1-r_i}}$$

Constant for each query

Only quantity to be estimated for rankings

- Retrieval Status Value:

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

Binary Independence Model

- All boils down to computing RSV.

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

$$RSV = \sum_{x_i = q_i = 1} c_i; \qquad c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

So, how do we compute c_i's from our data ?

Remember: $p_i = p(x_i = 1 \mid R, q); \quad r_i = p(x_i = 1 \mid NR, q);$

Binary Independence Model

- Estimating RSV coefficients.
- For each term i look at this table of document counts:

Documens	Relevant	Non-Relevant	Total
x_i=1	s	n-s	n
x_i=0	S-s	N-n-S+s	N-n
Total	S	N-S	N

- Estimates: $p_i \approx \frac{s}{S}$ $\quad r_i \approx \frac{(n-s)}{(N-S)}$

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

For now, assume no zero terms.

	T1	T2	T3	T4	T5	T6	Relevance
D1	1	0	1	1	0	0	R
D2	0	1	0	1	0	1	R
D3	1	0	1	1	1	0	NR
D4	0	1	1	0	1	1	NR
D5	1	1	0	1	0	0	NR
D6	1	1	0	1	1	1	NR
D7	0	0	0	0	0	1	R
D8	0	0	1	1	1	0	NR
D9	1	1	1	0	1	1	R
D10	1	0	0	1	1	0	NR

Suppose the relevance judgments specified above represent some past user judgments on the relevance of these documents wrt a given query. Using the basic probabilistic retrieval model, compute the discriminant (i.e., P(R|D) vs P(NR|D)) for each of the two new documents

D11 = (0,1,1,0,0,1)
D12 = (1,0,1,1,0,1)

with respect to the query Q = (1,1,0,1,0,1). Based on this discriminant, should these documents be retrieved? Explain your answer.

Estimation – Key Challenge

- If non-relevant documents are approximated by the whole collection, then r_i (prob. of occurrence in non-relevant documents for query) *is n/N* (where n=df_i) and

  $\log (1- r_i)/r_i = \log (N- n)/n \approx \log N/n = \text{IDF}$
- p_i (probability of occurrence in relevant documents) can be estimated in various ways:
  - from relevant documents, if we know some
    - Relevance weighting can be used in feedback loop
  - constant (Croft and Harper combination match) – each term has even odds of appearing in a relevant document - then just get idf weighting of terms
  - proportional to prob. of occurrence in collection
    - more accurately, to log of this (Greiff, SIGIR 1998)

Iteratively Estimating p_i

1. Assume that p_i constant over all x_i in query
   - p_i = 0.5 (even odds) for any given doc
2. Determine guess of relevant document set:
   - V is fixed size set of highest ranked documents on this model
3. We need to improve our guesses for p_i and r_i, so
   - Use distribution of x_i in docs in V. Let V_i be set of documents containing x_i
     - $p_i = |V_i| / |V|$
   - Assume if not retrieved, then not relevant
     - $r_i = (n_i - |V_i|) / (N - |V|)$
4. Go to 2. until converges, then return ranking

Probabilistic Relevance Feedback

1. Guess a preliminary probabilistic description of *R* and use it to retrieve a first set of documents V, as above.
2. Interact with the user to refine the description: learn some definite members of R and NR
3. Re-estimate p_i and r_i on the basis of these
   - Or can combine new information with original guess (use Bayesian prior):

$$p_i^{(2)} = \frac{|V_i| + \kappa p_i^{(1)}}{|V| + \kappa}$$

κ is prior weight

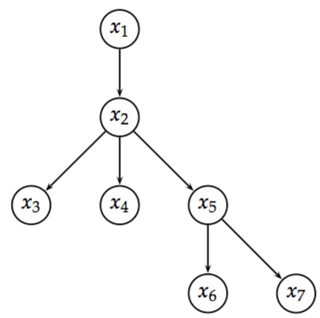
4. Repeat, thus generating a succession of approximations to *R*.

PRP and BIR

- Getting reasonable approximations of probabilities is possible.
- Requires restrictive assumptions:
  - ***term independence***
  - ***terms not in the query don't affect the outcome***
  - ***Boolean representation of documents/queries/relevance***
  - ***document relevance values are independent***
- Some of these assumptions can be removed
- Problem: either require partial relevance information or only can derive somewhat inferior term weights

Removing Term Independence

- In general, index terms aren't independent
- Dependencies can be complex
- van Rijsbergen (1979) proposed model of simple tree dependencies
  - Exactly Friedman and Goldszmidt' s Tree Augmented Naive Bayes (AAAI 13, 1996)
- Each term dependent on one other
- In 1970s, estimation problems held back success of this model



Good and Bad News

- Standard Vector Space Model
  - Empirical for the most part; success measured by results
  - Few properties provable
- Probabilistic Model
  - Advantages
    - Based on a firm theoretical foundation
    - Theoretically justified optimal ranking scheme
  - Disadvantages
    - Making the initial guess to get V
    - Binary word-in-doc weights (not using term frequencies)
    - Independence of terms (can be alleviated)
    - Amount of computation
    - Has never worked convincingly better in practice, but still an active research area