Relevance Feedback

Lecture 11

Query Operations

- Creating good search queries is hard
  - little knowledge of the collection
  - or retrieval system, query language, etc.
- Treat first query as just a first attempt to find relevant information
- Using the initial retrieved set, use
  - relevance information elicited from the user
  - statistics of terms in the retrieved set
  - statistics from the document collection

... to improve the query

Relevance Feedback

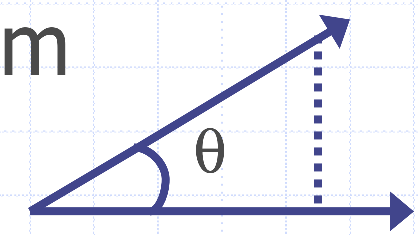
- In response to a ranked list, a user provides relevance information for some documents
- Two ways to improve a free-text query:
  1. Expanding the set of terms in the query
  2. Adjusting the weights of the query terms
- Goal of relevance feedback
  - add query terms and adjust term weights
  - improve ranks of known relevant documents
  - other relevant docs will also be ranked higher

The Vector Space Model

- Documents and queries are both vectors

$$\vec{d}_i = (w_{i,1}, w_{i,2} \ldots w_{i,t})$$

  - each $w_{i,j}$ is a weight for term j in document i
  - "bag-of-words representation"
- Similarity of a document vector to a query vector = cosine of the angle between them



Cosine Similarity Measure

$$sim(d_i, q) = \cos\theta$$

$$(x \cdot y = |x||y| \cos\theta)$$

$$= \frac{d_i \cdot q}{|d_i||q|} = \frac{\sum_j w_{i,j} \times w_{q,j}}{\sqrt{\sum_j w_{i,j}^2} \sqrt{\sum_j w_{q,j}^2}}$$

- Cosine is a normalized dot product
- Documents ranked by decreasing cosine value
  - sim(d,q) = 1 when d = q
  - sim(d,q) = 0 when d and q share no terms

Relevance Feedback in VSM

- In the vector space model, we assume
  1. relevant documents are similar to each other
  2. irrelevant documents are not similar to the relevant documents
  3. the best query is that which is closest to the relevant documents
- VSM relevance feedback strategy:
  - reformulate query vector so that it is closer to the space containing the relevant documents

Optimal Feedback Query

- Notation
  - N documents in collection
  - D: retrieved set, partitioned into D_r and D_n
  - C: collection, partitioned into C_r and C_n
  - $|D_r|$ = number of documents in D_r
- The optimal query maximizes similarity to relevant documents, and minimizes similarity to irrelevant ones
- If we know C_r in advance, the optimal query is given by:

$$\vec{q}_{\mathrm{opt}} = \frac{1}{C_r} \sum_{\vec{d} \in C_r} \vec{d} - \frac{1}{N - C_r} \sum_{\vec{d} \notin C_r} \vec{d}$$

VSM Relevance Feedback

- The optimal query is unknown
- The user can't specify C_r
- Solution: allow incremental changes to the query vector based on feedback
  - Reformulate query using *known* relevance information only

Rocchio's Formula

$$\vec{q}_{i+1} = \alpha\vec{q}_i + \frac{\beta}{|D_r|}\sum_{\vec{d}\in D_r}\vec{d} - \frac{\gamma}{|D_n|}\sum_{\vec{d}\in D_n}\vec{d}$$

- α, β, and γ give relative weight of q, D_r, and D_n
  - original query has crucial terms
  - relevant documents add new, useful terms
  - terms from D_r are more useful than terms from D_n
- β and γ-terms are weighted centroids of D_r and D_n
- q_{i+1} weights are normalized so that $q_{i+1} >= 0$

Ide's formulations (Salton 71)

$$\text{Ide}-\text{regular}: \vec{q}_{i+1} = \alpha\vec{q}_i + \beta\sum_{\vec{d}\in D_r}\vec{d} - \gamma\sum_{\vec{d}\in D_n}\vec{d}$$

$$\text{Ide}-\text{DecHi}: \vec{q}_{i+1} = \alpha\vec{q}_i + \beta\sum_{\vec{d}\in D_r}\vec{d} - \gamma\,\underset{\vec{d}\in \mathrm{D_n}}{\mathrm{argmax}}\, sim(\vec{d},q)$$

- Ide_regular does not explicitly normalize by the set sizes
- Ide_dec_hi only uses the highest-ranked known irrelevant document

Term Selection Problem

- Which terms should be included in the expanded query?
  1. Use original query terms only
  2. Use all terms contained in feedback documents
  3. Partial expansion
     - most common terms: highest within-doc freq.
     - highest weighted terms

Comparison to the Probabilistic Model

$$\mathrm{sim}_{\mathrm{PR}} = \sum_{i \in D} \log \frac{(r_i + 0.5)(N - R - n_i + r_i + 0.5)}{(n_i - r_i + 0.5)(R - r_i + 0.5)}$$

- In the probabilistic model
  - Relevance information is incorporated directly when computing term weights
  - Not possible to weight query and document terms differently
- Okapi model does provide these features

Relevance Feedback Evaluation

- Salton and Buckley (JASIS 90)
  - evaluated RF performance in 6 document collections
  - 12 methods (both VSM and probabilistic)
  - Rocchio weights: α=1, β=0.75, γ=0.25
- Experimental approach
  - Initial query weighted using tf*idf variant
  - Assessed top 15 documents retrieved using collection relevance judgments
  - Used resulting D_r and D_n to build feedback query
  - Measured improvement in 3-pt average precision

Salton and Buckley's Results

- Relevance feedback can greatly improve retrieval effectiveness
- Full expansion better than partial or none
  - improvement over partial is small
- Ide Dec-Hi performed best
  - followed closely by Rocchio
  - Probabilistic methods perform a little worse
- Greatest improvements seen when
  - the initial query is short, or
  - initial retrieval performance is poor