3 Relevance Feedback

3.1 Basics

Given: a query q, a result set (or ranked list) D,
a user‘s assessment u: D $\rightarrow$ {+, −}
yielding positive docs $D^+ \subseteq D$ and negative docs $D^- \subseteq D$

Goal: derive query q‘ that better captures the user‘s intention
or a better suited similarity function, e.g., by
- changing weights in the query vector or
- changing weights for different aspects of similarity
  (color vs. shape in multimedia IR, different colors,
  relevance vs. authority vs. recency)

Classical approach: ***Rocchio method*** (for term vectors)

$$q' = \alpha q + \frac{\beta}{|D^+|} \sum_{d \in D^+} d - \frac{\gamma}{|D^-|} \sum_{d \in D^-} d$$

with $\alpha, \beta, \gamma \in [0,1]$ and typically $\alpha > \beta > \gamma$

Generalized Query Point Movement

Assume the user rates (at least the positive) docs in result set D,
yielding feedback values rf(d) for $d \in D$
(e.g., 3=perfect match, 2=relevant doc, 1=ok if nothing better, etc.)

$$q' = \alpha q + \frac{\beta}{\sum_{d \in D^+} rf(d)} \sum_{d \in D^+} rf(d) \cdot d - \frac{\gamma}{\sum_{d \in D^-} rf(d)} \sum_{d \in D^-} rf(d) \cdot d$$

Pseudo-Relevance Feedback

based on J. Xu, W.B. Croft: Query expansion using local and
global document analysis, SIGIR Conference, 1996

Lazy users may perceive feedback as too bothersome

Evaluate query and simply view top n results as positive docs:
Add these results to the query and re-evaluate or
Select „best“ terms from these results and expand the query

3.2 Reshaping the Distance Measure

Assume that original distance measure (inverse similarity)
is a vector-space norm (e.g., Manhattan, Euclidean, etc.).
Use relevance feedback to adjust dimension weights:

$$L_p(x, y) = \sqrt[p]{\sum_{i=1}^{n} w_i \left(x_i - y_i\right)^p}$$

Choose weights w_i inversely proportional to
variance in dimension-i features of positive docs

$$w_i \sim 1/Var[d_i \mid d \in D^+]$$

Avoid „overshooting“:

$$w_i^{new} := \alpha \cdot w_i^{old} + \beta \cdot w_i$$

Adjusting Distances based on Quadratic Form

Consider distance function (Mahalanobis distance) with
n×n feature-feature similarity matrix M:

$$dist(x, y) = (x - y)^T M \; (x - y) = \sum_i \sum_j m_{ij} \cdot (x_i - y_i) \cdot (x_j - y_j)$$

Given feedback rf(d) for each d in D^+, determine M and q‘ such that:

$$\sum_{d \in D^+} rf(d) \cdot (d - q')^T M (d - q') = \min! \quad \text{and } \det(M) = 1$$

Optimal solution is:

$$q' = \left(\sum_{d \in D^+} rf(d) \cdot d\right) / \left(\sum_{d \in D^+} rf(d)\right)$$

$$M = \det(C)^{1/n} C^{-1} \quad \text{with} \quad C_{ij} = \sum_{d \in D^+} rf(d)(d_i - q_i)(d_j - q_j)$$

Adjusting Weights in Multi-Criteria Distance

Consider distance function with multiple, weighted criteria:

$$dist(d,q) = \sum_{k=1}^{m} w_k \cdot dist_k(d,q)$$

D+ (possibly over several queries) and $rf_q(d^{(i)})$ for $d^{(i)} \in$ D+ yields a set of sample points $(x_1^{(i)}, ..., x_m^{(i)}, y^{(i)})$ with $x_1^{(i)} = dist_1(d^{(i)},q), ..., x_m^{(i)} = dist_m(d^{(i)},q), y^{(i)} = rf_q(d^{(i)})$

„Learn" the optimal weights w_k by linear regression:
minimize the squared error

$$\sum_i \left(\left(\sum_k w_k x_k^{(i)} \right) - y^{(i)} \right)^2 =: E(w_1, ..., w_m)$$

Solve linear equation system: $\frac{\partial E}{\partial w_k} = 0$ for k=1, ..., m

Query Expansion: Adding Features

Generate single-feature query candidates c1, .., cm from D+,
e.g., extracting the best (tf or MI based) terms from positive docs

For each candidate ci, compute:

E[dist(ci,d) | d ∈D+] =: E+ (ci)
E[dist(ci,d) | d ∈D–] =: E– (ci)
Var[dist(ci,d) | d ∈D+] =: V+ (ci)
Var[dist(ci,d) | d ∈D–] =: V– (ci)

Consider adding ci to the query (i.e., setting q' = q + ci) if the ***separation distance*** is positive (and sufficiently high):

$sep(ci) = (E^{-}(ci) - V^{-}(ci)^{1/2}) - (E^{+}(ci) + V^{+}(ci)^{1/2})$

3.3 Profile Management

Long-term feedback obtained from many queries
of the same user or user group
may be captured in the form of a user profile,
which tracks *user-specific weights* and other feedback-based params

A profile may represent the union of positive docs
from earlier queries simply by the *centroid*.
When a user gives *feedback to a new query*, the most similar
profile is determined and the query is adjusted based on this profile.

Long-term profile management may involve
merging or splitting profiles.

Literature

- Michael Ortega-Binderberger, Sharad Mehrotra:
  Relevance Feedback in Multimedia Databases,
  in: Borko Furht and Oge Marquez (Editors),
  Handbook of Video Databases: Design and Applications,
  CRC Press, 2003
- Michael Ortega-Binderberger, Kaushik Chakrabarti, Sharad Mehrotra:
  An Approach to Integrating Query Refinement in SQL,
  EDBT Conference, 2002
- Yoshiharu Ishikawa, Ravishankar Subramanya, Christos Faloutsos:
  MindReader: Querying Databases Through Multiple Examples,
  VLDB Conference, 1998
- Ugur Cetintemel, Michael Franklin, Lee Giles: Self-adaptive User
  Profiles for Large-scale Data Delivery, ICDE Conference, 2000