Introduction to

Information Retrieval

Relevance Feedback

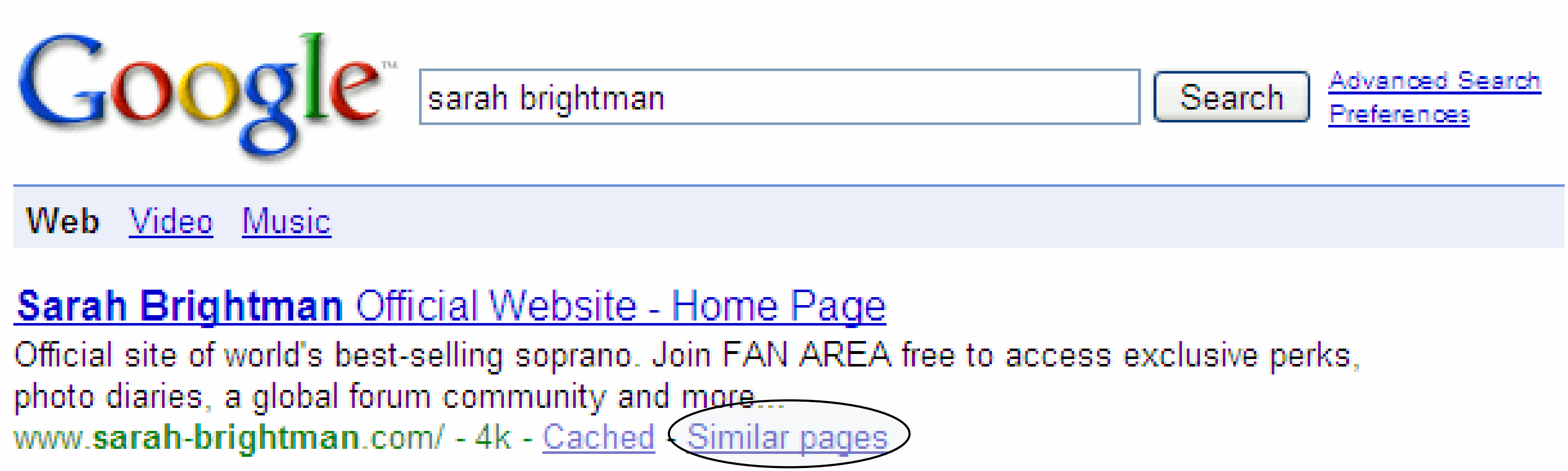
Relevance Feedback

- Relevance feedback: user feedback on relevance of docs in initial set of results
  - User issues a (short, simple) query
  - The user marks some results as relevant or non-relevant.
  - The system computes a better representation of the information need based on feedback.
  - Relevance feedback can go through one or more iterations.
- Idea: it may be difficult to formulate a good query when you don't know the collection well, so iterate

Relevance feedback

- We will use *ad hoc retrieval* to refer to regular retrieval without relevance feedback.
- We now look at four examples of relevance feedback that highlight different aspects.

Similar pages



sarah brightman

Search

Advanced Search
Preferences

Web Video Music

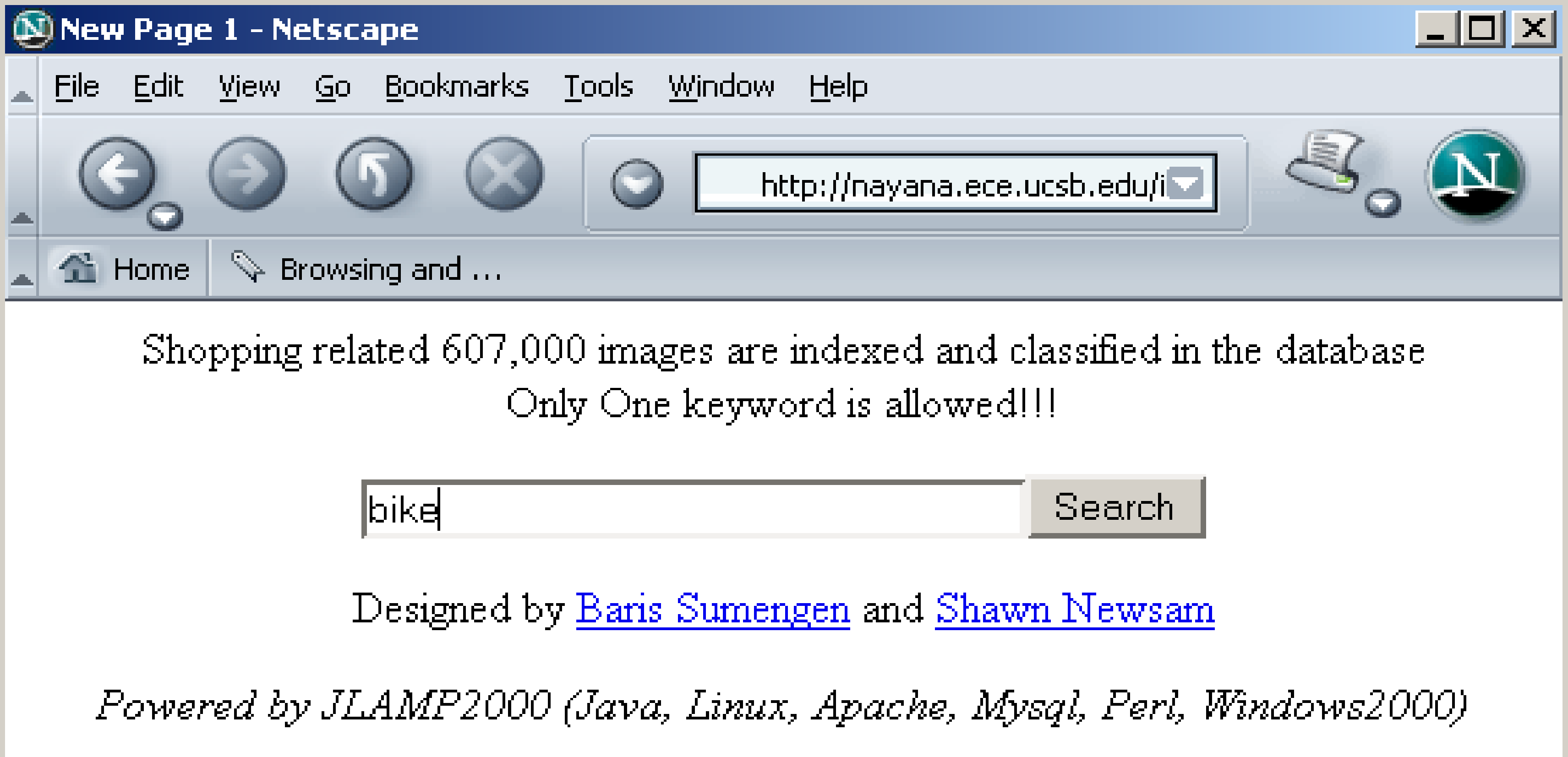
Sarah Brightman Official Website - Home Page

Official site of world's best-selling soprano. Join FAN AREA free to access exclusive perks, photo diaries, a global forum community and more...

www.sarah-brightman.com/ - 4k - Cached - Similar pages

Relevance Feedback: Example

- Image search engine
  http://nayana.ece.ucsb.edu/imsearch/imsearch.html


New Page 1 - Netscape
File Edit View Go Bookmarks Tools Window Help
http://nayana.ece.ucsb.edu/i
Home
Browsing and ...

Shopping related 607,000 images are indexed and classified in the database
Only One keyword is allowed!!!

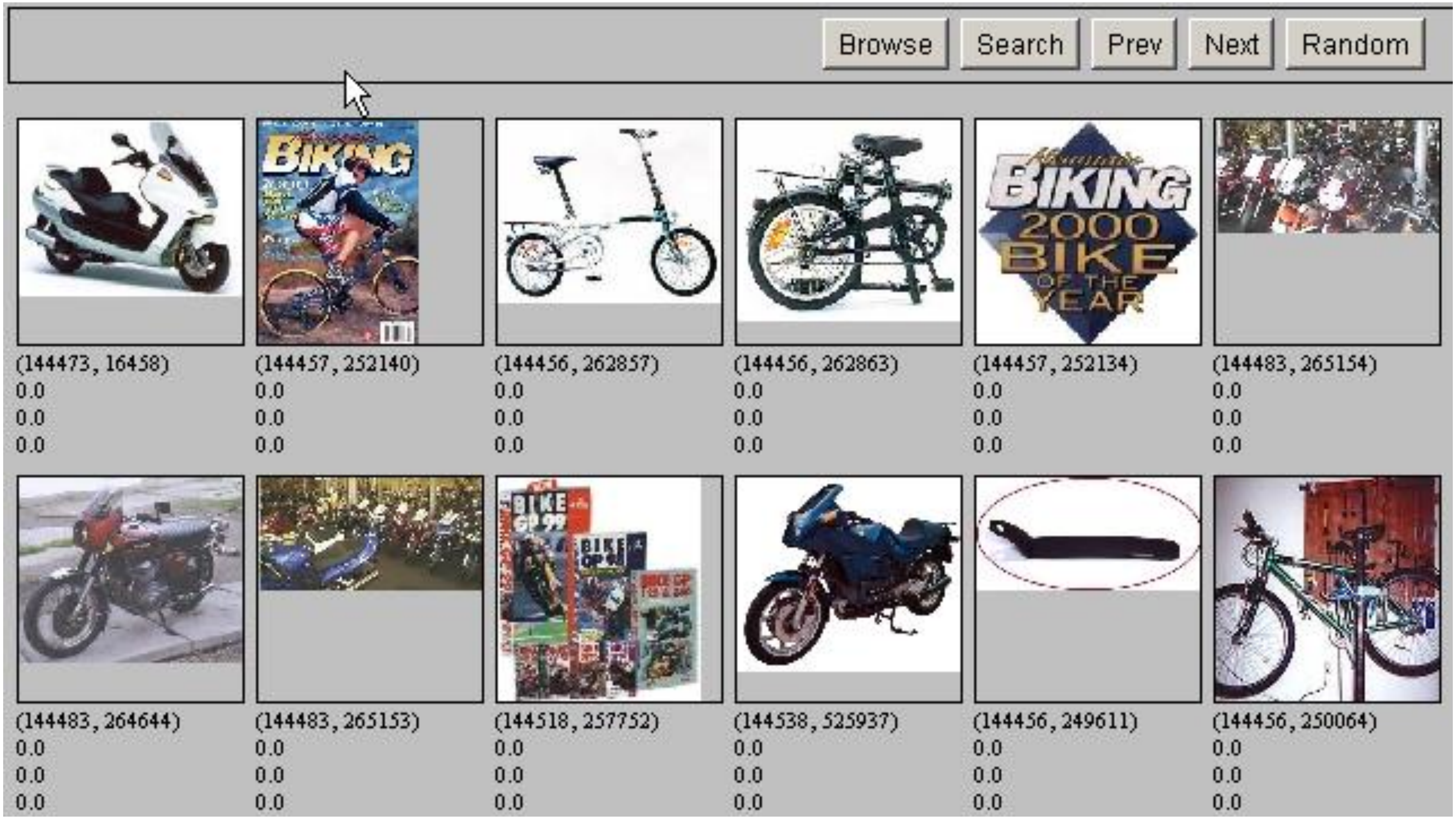
bike

Search

Designed by Baris Sumengen and Shawn Newsam

Powered by JLAMP2000 (Java, Linux, Apache, Mysql, Perl, Windows2000)

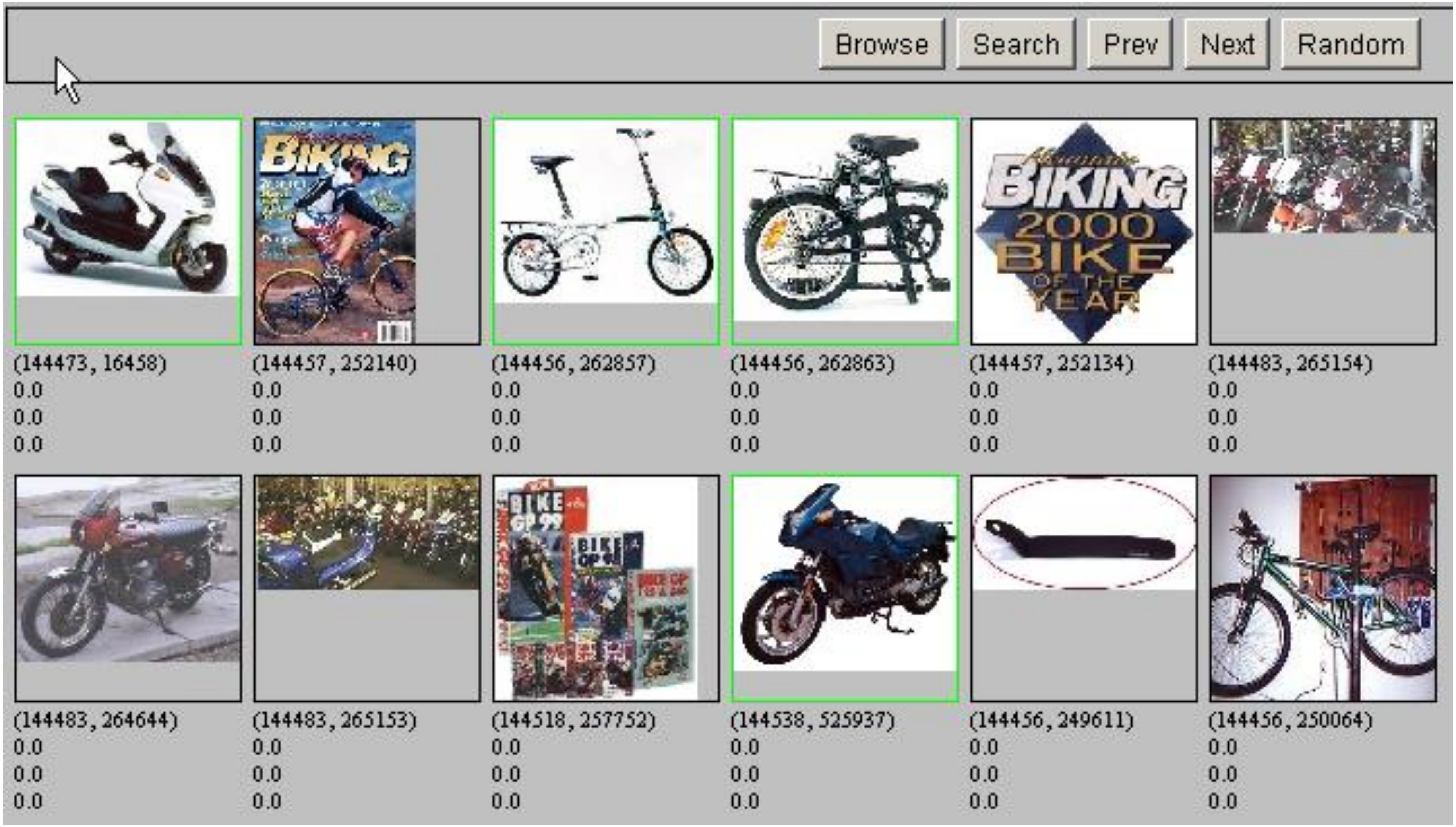
Results for Initial Query



```
(144473, 16458)   (144457, 252140)   (144456, 262857)   (144456, 262863)   (144457, 252134)   (144483, 265154)
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0

(144483, 264644)  (144483, 265153)   (144518, 257752)   (144538, 525937)   (144456, 249611)   (144456, 250064)
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0
```

Relevance Feedback



(144473, 16458)
0.0
0.0
0.0

(144457, 252140)
0.0
0.0
0.0

(144456, 262857)
0.0
0.0
0.0

(144456, 262863)
0.0
0.0
0.0

(144457, 252134)
0.0
0.0
0.0

(144483, 265154)
0.0
0.0
0.0

(144483, 264644)
0.0
0.0
0.0

(144483, 265153)
0.0
0.0
0.0

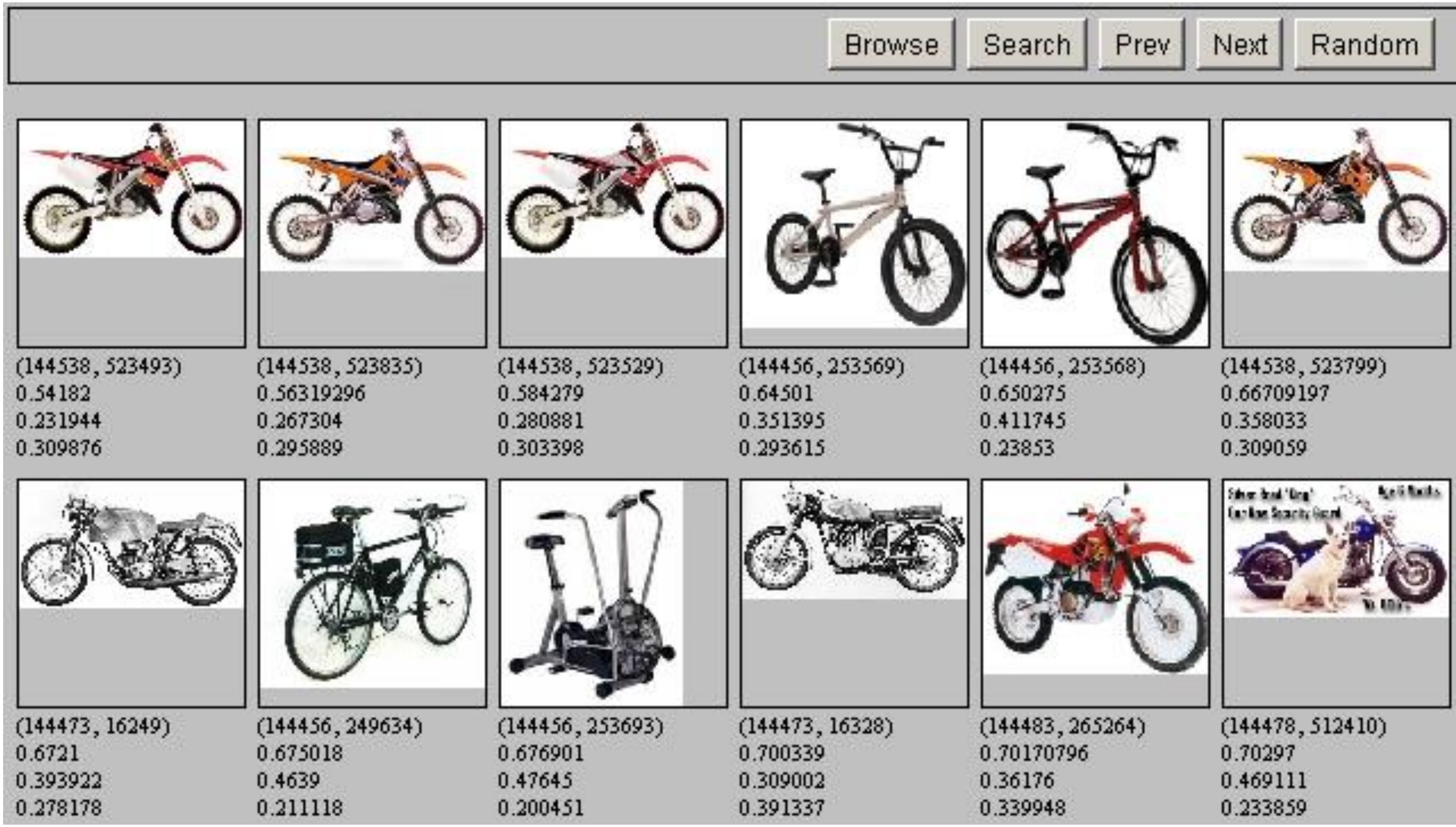
(144518, 257752)
0.0
0.0
0.0

(144538, 525937)
0.0
0.0
0.0

(144456, 249611)
0.0
0.0
0.0

(144456, 250064)
0.0
0.0
0.0

Results after Relevance Feedback



(144538, 523493)
0.54182
0.231944
0.309876

(144538, 523835)
0.56319296
0.267304
0.295889

(144538, 523529)
0.584279
0.280881
0.303398

(144456, 253569)
0.64501
0.351395
0.293615

(144456, 253568)
0.650275
0.411745
0.23853

(144538, 523799)
0.66709197
0.358033
0.309059

(144473, 16249)
0.6721
0.393922
0.278178

(144456, 249634)
0.675018
0.4639
0.211118

(144456, 253693)
0.676901
0.47645
0.200451

(144473, 16328)
0.700339
0.309002
0.391337

(144483, 265264)
0.70170796
0.36176
0.339948

(144478, 512410)
0.70297
0.469111
0.233859

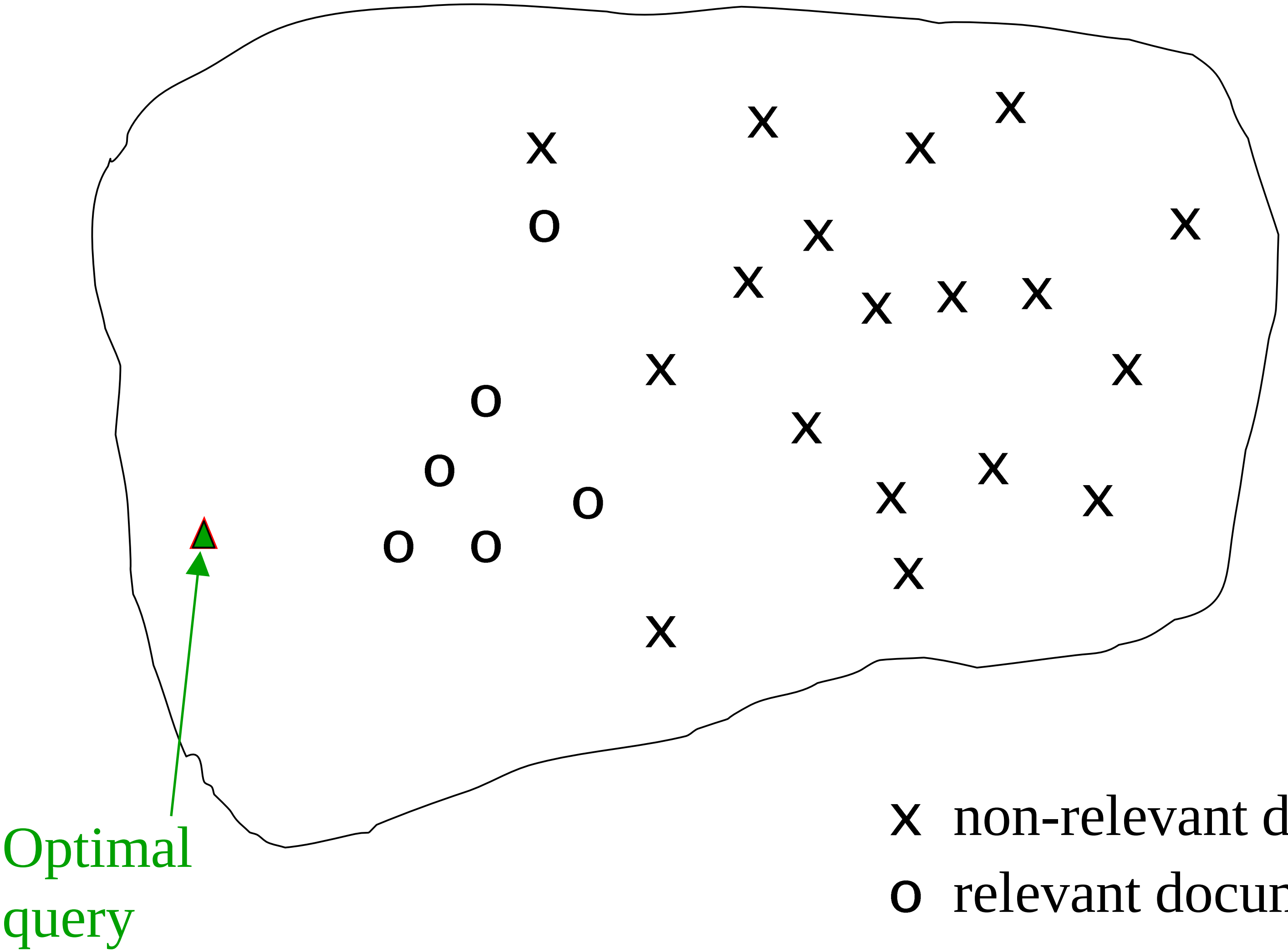
Key concept: Centroid

- The centroid is the center of mass of a set of points
- Recall that we represent documents as points in a high-dimensional space
- Definition: Centroid

$$\vec{\mu}(C) = \frac{1}{|C|}\sum_{d \in C} \vec{d}$$

where C is a set of documents.

The Theoretically Best Query



x non-relevant documents

o relevant documents

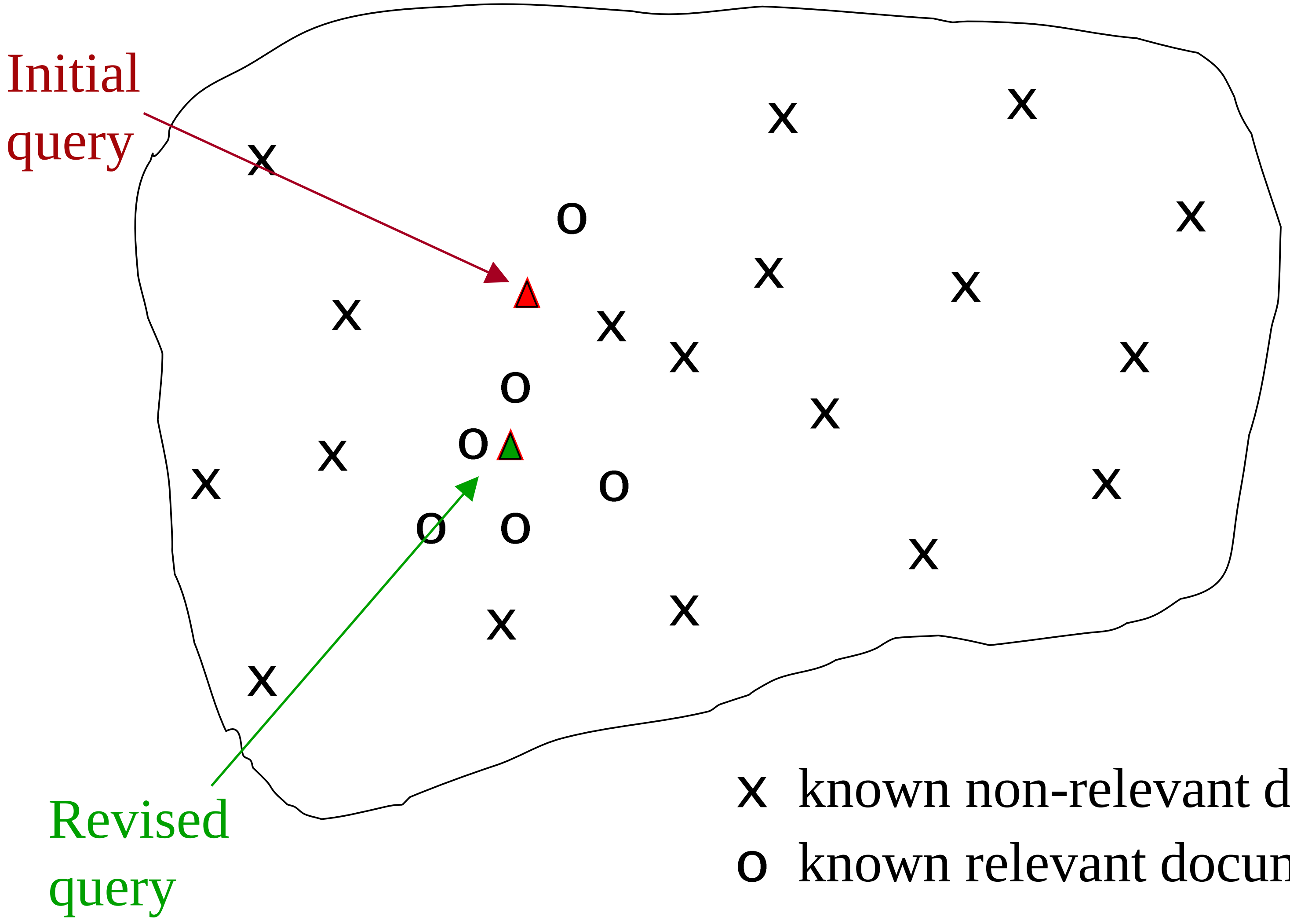
Rocchio 1971 Algorithm

- Used in practice:

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}}\vec{d}_j$$

- D_r = set of <u>known</u> relevant doc vectors
- D_{nr} = set of <u>known</u> irrelevant doc vectors
- q_m = modified query vector; q_0 = original query vector; α,β,γ: weights (hand-chosen or set empirically)(typical values **a = 1**, **b = 0.8**, and **ϒ = 0.1**.
- New query moves toward relevant documents and away from irrelevant documents

Relevance feedback on initial query



x known non-relevant documents

o known relevant documents

Relevance Feedback in vector spaces

- We can modify the query based on relevance feedback and apply standard vector space model.
- Use only the docs that were marked.
- Relevance feedback can improve recall and precision
- Relevance feedback is most useful for increasing *recall* in situations where recall is important
  - Users can be expected to review results and to take time to iterate

Positive vs Negative Feedback

- Positive feedback is more valuable than negative feedback (so, set $\gamma < \beta$; e.g. $\gamma = 0.25$, $\beta = 0.75$).
- Many systems only allow positive feedback (γ=0).

Relevance Feedback: Assumptions

- A1: User has sufficient knowledge for initial query.
- A2: Relevance prototypes are “well-behaved”.
  - Term distribution in relevant documents will be similar
  - Term distribution in non-relevant documents will be different from those in relevant documents
    - Either: All relevant documents are tightly clustered around a single prototype.
    - Or: There are different prototypes, but they have significant vocabulary overlap.
    - Similarities between relevant and irrelevant documents are small

Violation of A1

- User does not have sufficient initial knowledge.
- Examples:
  - Misspellings (Brittany Speers).
  - Cross-language information retrieval (hígado).
  - Mismatch of searcher's vocabulary vs. collection vocabulary
    - Cosmonaut/astronaut

Violation of A2

- There are several relevance prototypes.
- Examples:
  - Burma/Myanmar
  - Contradictory government policies
  - Pop stars that worked at Burger King
- Often: instances of a general concept
- Good editorial content can address problem
  - Report on contradictory government policies

Relevance Feedback: Problems

- Long queries are inefficient for typical IR engine.
  - Long response times for user.
  - High cost for retrieval system.
  - Partial solution:
    - Only reweight certain prominent terms
      - Perhaps top 20 by term frequency
- Users are often reluctant to provide explicit feedback
- It's often harder to understand why a particular document was retrieved after applying relevance feedback

Relevance Feedback on the Web

- Some search engines offer a similar/related pages feature (this is a trivial form of relevance feedback)
  - Google (link-based)
  - Altavista
  - Stanford WebBase

α/β/γ ??

- But some don’t because it’s hard to explain to average user:
  - Alltheweb
  - bing
  - Yahoo
- Excite initially had true relevance feedback, but abandoned it due to lack of use.

Excite Relevance Feedback

Spink et al. 2000

- Only about 4% of query sessions from a user used relevance feedback option
  - Expressed as “More like this” link next to each result
- But about 70% of users only looked at first page of results and didn’t pursue things further
  - So 4% is about 1/8 of people extending search
- Relevance feedback improved results about 2/3 of the time

Pseudo relevance feedback

- Pseudo-relevance feedback automates the “manual” part of true relevance feedback.
- Pseudo-relevance algorithm:
  - Retrieve a ranked list of hits for the user’s query
  - Assume that the top k documents are relevant.
  - Do relevance feedback (e.g., Rocchio)
- Works very well on average
- But can go horribly wrong for some queries.
- Several iterations can cause query drift.
- Why?