Heaps' law

$$M = kT^b$$

M = distinct terms in a collection
k = constant 30 ≤ k ≤ 100
T = tokens in the collection
b = constant ≈ 0.5

The relationship between vocabulary size (M) and collection size (T) is linear in log-log space.

Zipf's law

$$cf_i \propto \frac{1}{i}$$

The collection frequency cf_i of the ith most common term is proportional to 1 / i

Bayes theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Tf-idf

$$tfidf_{t,d} = tf_{t,d} \times idf_t$$

$$idf_t = \log \frac{N}{df_t}$$

$tf_{t,d}$ = Term frequency, in the course term count in d
df_t = Document frequency, in the course # of documents with the term
N = Total # of documents in collection

Cosine similarity

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| \times |\vec{V}(d_2)|}$$

Cosine similarity measures similarity as the angle between the two document vectors which is the dot product of two normalized vectors.

Rocchio relevance feedback

$$\vec{q}_m = a\vec{q}_0 + b\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r} \vec{d}_j - c\frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

D_r = set of relevant documents
D_{nr} = set of non-relevant documents
a, b, c = weighting values

Multinomial naive Bayes

$$P(c \mid d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k \mid c)$$

The probability of class c given document d is proportional to the general probability of the class times the product of probabilities of terms in the document being in that class.

Add-one-smoothing

$$P(t|c) = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B'}$$

T_{ct} = Term count of t in training documents of the class
B' = $|V|$ = Size of vocabulary
$\sum_{t' \in V} T_{ct'}$ = Size of text in training documents of the class

Rocchio classifier

Centroid of a class c:

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

I.e. the average of the document vectors. A document is classified depending on which centroid is closest.

kNN - k nearest neighbor

Looking at the k nearest neighbors to the document, classify the document according to the most common among the chosen neighbors.

Support vector machines

$$f(\vec{x}) = \text{sign}(\vec{w}^T \vec{x} + b)$$

$\vec{x}$ = Document to classify
$\vec{w}^T$ = Normal vector, transposed
b = Offset (from optimization problem)

$$\vec{w} = \sum a_i y_i \vec{x}_i$$

a_i = Lagrange multiplier (from optimization problem)
y_i = Class of document $\vec{x}_i$, -1 or 1

Precision / recall

Precision (P) is the fraction of retrieved documents that are relevant
Recall (R) is the fraction of relevant documents that are retrieved

Precision = # of relevant results / # of results = P(relevant | retrieved)
Recall = # of relevant results / # of relevant items = P(retrieved | relevant)

$$F_1 = F_{\beta=1} = \frac{2PR}{P+R}$$

Jaccard coefficient

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

NDCG

Normalized discounted cumulative gain

$$\text{CG}_p = \sum_{i=1}^{p} r_i$$

r_i = graded relevance of result i

$$\mathrm{DCG}_p = r_1 + \sum_{i=2}^{p} \frac{r_i}{log_2 i}$$

$$\mathrm{NDCG}_p = \frac{\mathrm{DCG}_p}{\mathrm{IDCG}_p}$$

IDCG_p = Ideal DCG. DCG if results were perfectly sorted.