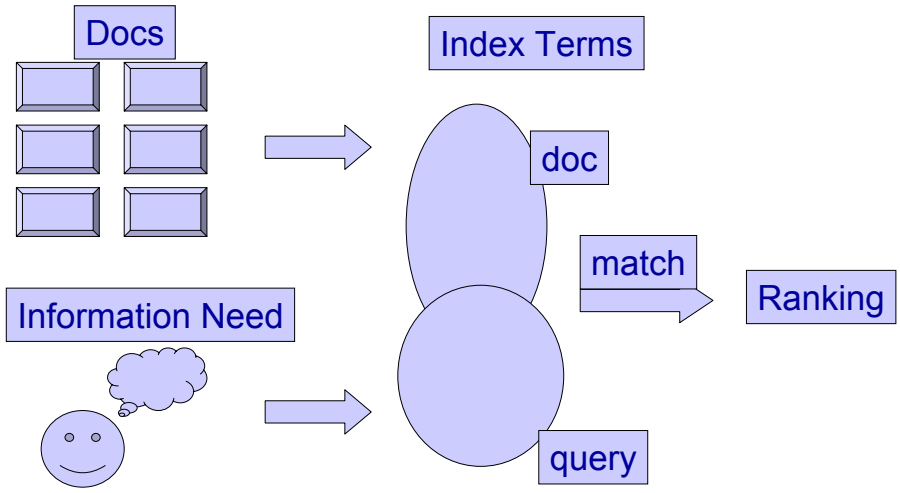
Modeling Issues in IR

Introduction

- IR systems usually adopt index terms to process queries
- Index terms may be:
  - a keyword or group of selected words, or
  - any word (more general)
- Stemming might be used, e.g.,
  - Connect could “replace”: connecting, connection and connections, both in the index as well as in the query
- An access structure/index (e.g., an inverted file) is built for the chosen index terms

Introduction



Introduction

- Matching at index term level is quite imprecise
- Not surprisingly users are, more often than not, not satisfied with the obtained answers
- Since most users have no training in query formation, problem becomes even worse, specially in a WWW environment
- Indeed, precise query formation is often not feasible and browsing is used to refine the original query
- Issue of deciding relevance is critical for IR systems (ranking)

Introduction

- *Ranking* is the ordering of the documents retrieved reflecting (hopefully) their relevance to the query
- A rank is based on fundamental premisses regarding the notion of relevance, e.g., common sets of index terms, sharing of weighted terms and likelihood of relevance
- Each set of premisses leads to a distinct *IR model*

Classic IR Models

- Each document represented by a set of representative keywords or index terms
- An index term is (usually) a document word useful for "remembering" the document main themes
- Usually, index terms are nouns because nouns have meaning by themselves
- However, search engines assume that almost all words are index terms (full text representation). A large set of words, the stopwords (e.g., a, an, the, etc), are not indexed

Classic IR Models

- Not all terms are equally useful for representing the contensts of a document. Less frequent terms allow identifying a narrower set of documents
- The *importance* of an index terms is represented by weights associated to it
- Let
  - k_i be an index term
  - d_j be a document
  - w_{ij} is a weight associated with (k_i, d_j)
- The weight w_{ij} quantifies the importance of the index term k_i for describing document d_j contents

Classic IR Models

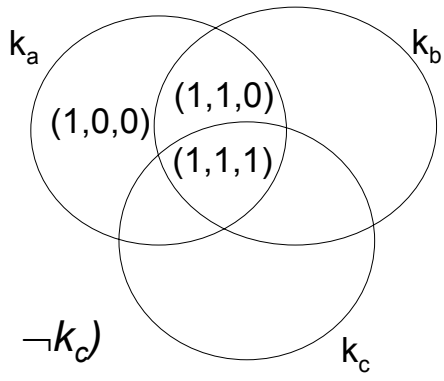
- k_i is an index term, i = 1, 2, ..., t
- d_j is a document, j = 1, 2, ... D
- $K = (k_1, k_2, \ldots, k_t)$ is the set of all index terms
- $w_{ij} >= 0$ is the weight associated with (k_i, d_j)
- $w_{ij} = 0$ indicates that term k_i does not help describing d_j
- $\vec{d}_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$ is a vector associated with (describing the) document d_j
- $gi(\vec{d}_j) = w_{ij}$ is a function which returns the weight associated with pair (k_i, d_j)

The Boolean Model

- Simple model based on set theory
- Queries specified as boolean expressions
  - precise semantics
  - elegant formalism
  - $q = k_a$ AND $(k_b$ OR $\neg k_c)$
- Terms are either present or absent.
- Thus, w_{ij} = 0 or 1

The Boolean Model

Consider



$q = k_a$ AND $(k_b$ OR $\neg k_c)$

$\vec{q}_{dnf}$ = *(1,1,1)* OR *(1,1,0)* OR *(1,0,0)*

Is the above query's disjunctive normal form

$\vec{q}_{cc}$ = *(1,1,0)* is a conjunctive component

The Boolean Model

$$sim(q, d_j) = \begin{cases} 1 \; if \;\; \exists \vec{q}_{cc} \mid \vec{q}_{cc} \in \vec{q}_{dnf} \; AND \, (\forall k_i \; g_i(\vec{d}_j) = g_i(\vec{q}_{cc})) \\ 0 \; otherwise \end{cases}$$

Note that according to the above definition a document d_j *for which* $\vec{d}_j$ = (0, 1, 0) would be deemed non-similar thus not retrieved, even though it could be considered partially relevant as it would contain k_b

Drawbacks of the Boolean Model

- Retrieval based on binary decision criteria with no notion of partial matching
- No ranking of the documents is provided
- Information need has to be translated into a Boolean expression which most users find awkward
- The Boolean queries formulated by the users are most often too simplistic – thus the model frequently returns either too few or too many documents in response to a user query

The Vector Model

- Use of binary weights is too limiting
- Non-binary weights allow consideration of partial matches
- These term weights are used to compute a *degree of similarity* between a query and each document
- Ranked set of documents provides for better matching

The Vector Model

- Define:

  $w_{ij} > 0$ whenever $k_i \in d_j$

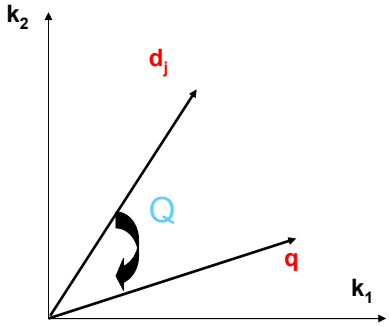
  $w_{iq} >= 0$ associated with pair (k_i, q) (q is the query)

  $\vec{d}_j = (w_{1j}, w_{2j}, ..., w_{tj})$

  $\vec{q} = (w_{1q}, w_{2q}, ..., w_{tq})$
- A unitary vector $\vec{i}$ is associated to each term k_i
- These unitary vectors are assumed to be orthonormal (i.e., index terms are assumed to occur independently within the documents)
- The *t* unitary vectors $\vec{i}$ form an orthonormal basis for a t-dimensional space
- In this space, queries and documents are represented as weighted vectors

The Vector Model

$$Sim(q,d_j) = \cos(Q) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j||\vec{q}|} = \frac{\sum_{i=1}^{t} w_{ij} * w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} * \sqrt{\sum_{i=1}^{t} w_{iq}^2}}$$



Since $w_{ij} > 0$ and $w_{iq} > 0$, $0 <= sim(q,d_j) <= 1$. *Most importantly, a* document is retrieved even if it matches the query terms only partially

The Vector Model

$$Sim(q,d_j) = \frac{\sum_{i=1}^{t} w_{ij} * w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} * \sqrt{\sum_{i=1}^{t} w_{iq}^2}}$$

- How to compute the weights w_{ij} and w_{iq} ?
- A good weighting must take into account two effects:
  - quantification of intra-document contents (similarity): *tf* factor, the *term frequency* within a document
  - quantification of inter-documents separation (dissimilarity): *idf* factor, the *inverse document frequency*

$$w_{ij} = tf(i,j) * idf(i)$$

Zipf's Law

- Term frequency distributions tend to follow Zipf's law

$$frequency \times rank \approx k \approx 0.1$$

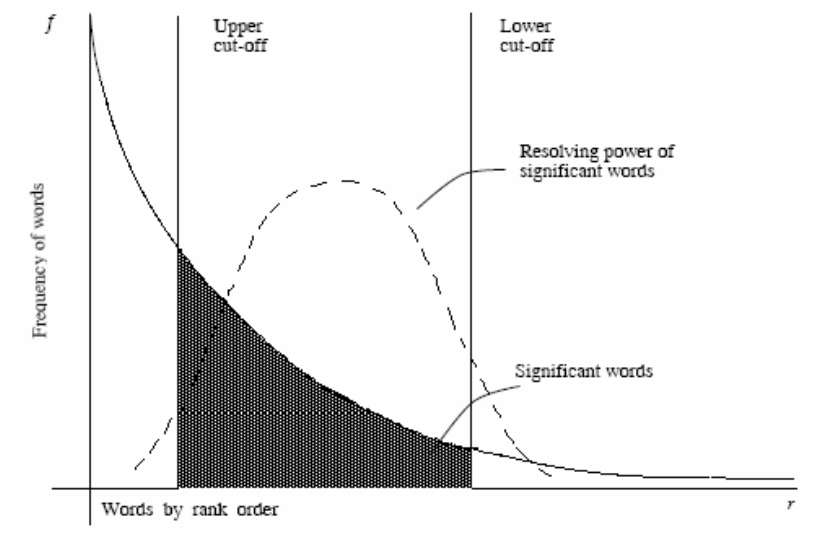
(Value of k observed for English)

• Another way to state this is with an approximately correct rule of thumb:

- Say the most common term occurs C times
- The second most common occurs C/2 times
- The third most common occurs C/3 times

…

Zipf's Law



The Vector Model

- N is the total number of docs in the collection
- n_i is the number of docs which contain *ki*
- *freq(i,j)* raw frequency of k_i within d_j
- A normalized *tf* factor is given by
  f(i,j) = freq(i,j) / max(freq(l,j))
  - where the maximum is computed over all terms which occur within the document d_j
- The *idf* factor is computed as
  idf(i) = log (N/n_i)
  - the *log* is used to make the values of *tf* and *idf* comparable. It can also be interpreted as the *amount of information* associated with the term k_i.

Inverse Document Frequency

- IDF provides high values for rare words and low values for common words

$$\log\left(\frac{10000}{10000}\right) = 0$$

$$\log\left(\frac{10000}{5000}\right) = 0.301$$

$$\log\left(\frac{10000}{20}\right) = 2.698$$

$$\log\left(\frac{10000}{1}\right) = 4$$

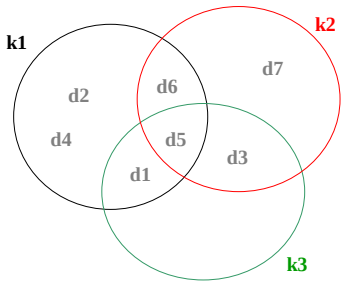
The Vector Model

- The so-called *tf-idf* weighting scheme use weights which are given by
  $w_{ij} = f(i,j) * log(N/n_i)$
- For the query term weights, a suggestion is
  $w_{iq} = (0.5 + [0.5 * freq(i,q) / max(freq(l,q)]) * log(N/n_i)$
- The vector model with *tf-idf* weights is a good (almost standard) ranking strategy with general collections. It is also simple and fast to compute.

The Vector Model

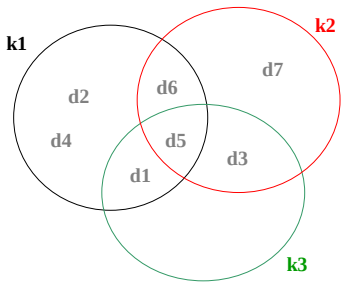
- Advantages:
  - term-weighting improves quality of the answer set
  - partial matching allows retrieval of docs that approximate the query conditions
  - cosine ranking formula sorts documents according to degree of similarity to the query
- Disadvantages:
  - assumes independence of index terms; not clear that this is bad though
  - Lacks the control of a Boolean model (e.g., requiring a term to appear in a document).
    - Given a two-term query “A B”, may prefer a document containing A frequently but not B, over a document that contains both A and B, but both less frequently.

The Vector Model: Example I



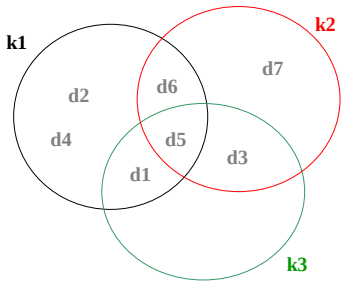
	k1	k2	k3	q • dj/|q||dj|
d1	1	0	1	**0,82**
d2	1	0	0	**0,58**
d3	0	1	1	**0,82**
d4	1	0	0	**0,58**
d5	1	1	1	**1**
d6	1	1	0	**0,82**
d7	0	1	0	**0,58**
q	1	1	1	

The Vector Model: Example II



	k1	k2	k3	q • dj/|q||dj|
d1	1	0	1	**0,76**
d2	1	0	0	**0,27**
d3	0	1	1	**0,94**
d4	1	0	0	**0,27**
d5	1	1	1	**0,93**
d6	1	1	0	**0,57**
d7	0	1	0	**0,53**
q	1	2	3	

The Vector Model: Example III



	k1	k2	k3	q • dj/|q||dj|
d1	2	0	1	**0,6**
d2	1	0	0	**0,27**
d3	0	1	3	**0,93**
d4	2	0	0	**0,27**
d5	1	2	4	**0,99**
d6	1	2	0	**0,6**
d7	0	5	0	**0,53**
q	1	2	3	

Set Theoretic Models

- The Boolean model imposes a binary criterion for deciding relevance
- The question of how to extend the Boolean model to accomodate partial matching and a ranking has attracted considerable attention in the past
- We discuss now two set theoretic models for this:
  - Fuzzy Set Model
  - Extended Boolean Model

Fuzzy Set Theory

- Framework for representing classes whose boundaries are not well defined
- Key idea is to introduce the notion of a *degree of membership* associated with the elements of a set
- This degree of membership varies from 0 to 1 and allows modeling the notion of *marginal* membership
- Thus, membership is now a *gradual* notion, contrary to the crispy notion enforced by classic Boolean logic

Κλασσικά σύνολα

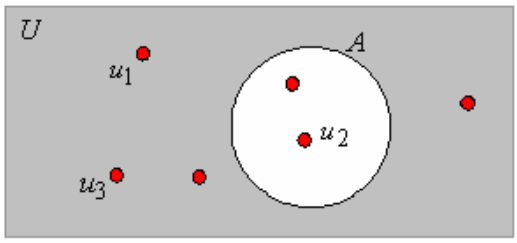
Θεωρούμε το σύνολο U των μαθητών μιά τάξης $U = (u_1, u_2, \cdots, u_N)$.

Εστω A είναι ένα υποσύνολο - κλάση του συνόλου U το οποίο περιλαμβάνει τους μαθητές με ύψος h πάνω από 160 cm:

$$A = \left(u_j,\ j \in [1, N]:\ h(u_j) > 160\,\text{cm}\right)$$

Τώρα, κάθε στοιχείο του συνόλου U έχει 2 δυνατότητες: ή συμμετέχει πλήρως στην κλάση A ή δεν συμμετέχει καθόλου.

Ας φανταστούμε μιά συνάρτηση μ_A η οποία δίδει σε κάθε στοιχείο u του συνόλου U το βαθμό συμμετοχής που έχει το στοιχείο αυτό στην κλάση A.



Από το σχήμα φαίνεται πως τα στοιχεία u_1 και u_3 δεν συμμετέχουν καθόλου στην κλάση A, ενώ το στοιχείο u_2 συμμετέχει πλήρως, Αρα η συνάρτηση μ_A γιά τα 3 αυτά στοιχεία θα έδινε βαθμό συμμετοχής:

$$\mu_A(u_1) = \mu_A(u_3) = 0 \quad \text{και} \quad \mu_A(u_2) = 1$$

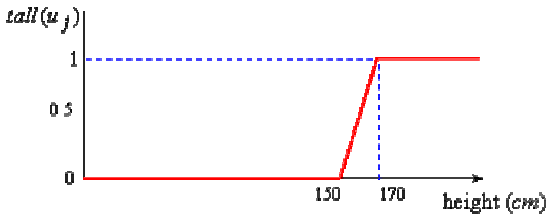
Fuzzy συνολα

Θεωρούμε πάλι το σύνολο U των μαθητών μιά τάξης: $U = (u_1, u_2, \cdots, u_N)$.

Ορίζουμε μιά κλάση A του συνόλου U η οποία θα περιέχει τους ψηλούς μαθητές της τάξης!!!

Πώς θα καθορίσουμε τους ψηλούς; Ας χρησιμοποιήσουμε ως κριτήριο την σχέση

$$\mu_A(u_j) \equiv tall(u_j) = \begin{cases} 0, & \text{αν } h(u_j) < 150\,cm \\ \dfrac{h(u_j) - 150}{20}, & \text{αν } 150\,cm \le h(u_j) \le 170\,cm \\ 1, & \text{αν } h(u_j) > 170\,cm \end{cases}$$



i	u_i	$h(u_i)$	$tall(u_i)$
1	Bob	161.5	0.575
2	Drew	175.2	1
3	Erik	147.8	0
4	Billy	168	0.9
5	Mark	155.4	0.27
≈	≈	≈	≈

Ας δούμε τώρα μαζί τα σύνολα U και A:

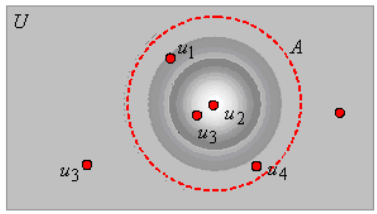
$$U = (u_1, \quad u_2, \quad u_3, \quad u_4, \quad u_5, \quad \cdots \quad u_N)$$

$$A = (0.575, \quad 1, \quad 0, \quad 0.9, \quad 0.27 \quad \cdots \quad tall(u_N))$$

Βλέπουμε ότι η κλάση A περιέχει, γιά όλα τα στοιχεία του συνόλου U, τις αντίστοιχες συμμετοχές στην έννοια "ψηλός". Βλέπουμε ότι η έννοια του ανήκειν στην κλάση δεν είναι απόλυτη όπως στα κλασσικά σύνολα. Εδώ η αλήθεια δεν είναι δυαδική {0, 1} αλλά υπάρχουν βαθμοί αλήθειας – συμμετοχής στην κλάση.

Γιά παράδειγμα, η φράση "ο Bob ειναι ψηλός" είναι αληθής κατά 0.575.

Η κλάση A απεικονίζεται στο σχήμα με ασαφή όρια που διαχέονται εξωτερικά εντός του συνόλου U:



Ασαφής θεωρία συνόλων

Η *ασαφής θεωρία συνόλων* (fuzzy set theory), ασχολείται με την αναπαράσταση των συνόλων, των οποίων τα όρια δεν είναι καλά ορισμένα.

Η βασική ιδέα είναι να βρούμε μιά συνάρτηση η οποία, σε κάθε στοιχείο ενός συνόλου, θα δίνει μιά τιμή μεταξύ 0 και 1.

ορισμός

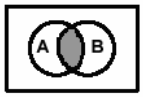
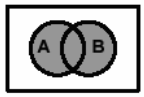
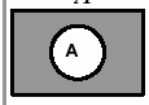
Ένα ασαφές υποσύνολο A ενός υπερσυνόλου U, χαρακτηρίζεται από μιά συνάρτηση συμμετοχής (membership function) $\mu_A : U \to [0,1]$ η οποία αντιστοιχεί σε κάθε στοιχείο u του U, έναν αριθμό $\mu_A(u)$ στο διάστημα $[0,1]$.

- ✓ Αν ένα στοιχείο u του συνόλου U, δίνει στην συνάρτηση συμμετοχής την τιμή 0, δεν συμμετέχει καθόλου στην κλάση A ενώ, αν δίνει τιμή 1, συμμετέχει πλήρως.
- ✓ Τιμές του βαθμού συμμετοχής, στο ενδιάμεσο διάστημα, προσδιορίζουν τα *οριακά στοιχεία* του συνόλου, δηλαδή στοιχεία για τα οποία υπάρχει αβεβαιότητα για το αν ανήκουν στο σύνολο. Η αβεβαιότητα μεγαλώνει όσο πιο μικρός είναι ο βαθμός συμμετοχής στο σύνολο.
- ✓ Με άλλα λόγια η συμμετοχή σε ένα ασαφές σύνολο έχει διάφορες *διαβαθμίσεις* και όχι διακριτό χαρακτήρα όπως στη Boolean λογική.

Πράξεις μεταξύ ασαφών συνόλων

Οι τρεις πιό κοινές πράξεις μεταξύ ασαφών συνόλων είναι

- Το **συμπλήρωμα** (complement) ενός ασαφούς συνόλου
- Η **ένωση** (union) δύο ή περισσοτέρων ασαφών συνόλων
- Η **τομή** (intersection) δύο ή περισσοτέρων ασαφών συνόλων

Intersection	Union	Complement
$A \cap B$	$A \cup B$	$\overline{A}$
		
$\mu_{A\cap B}(x) =$	$\mu_{A\cup B}(x) =$	$\mu_{\overline{A}}(x) =$
classical $\begin{cases} 1 & x \in A \cap B \\ 0 & x \notin A \cap B \end{cases}$	$\begin{cases} 1 & x \in A \cup B \\ 0 & x \notin A \cup B \end{cases}$	$\begin{cases} 1 & x \notin A \\ 0 & x \in A \end{cases}$
fuzzy $\min(\mu_A(x), \mu_B(x))$	$\max(\mu_A(x), \mu_B(x))$	$1 - \mu_A(x)$
AND	**OR**	**NOT**

- Εναλλακτικά:
- $\mu_{A \cup B}(x)=\mu_A(x) + \mu_B(x) - \mu_A(x)\ \mu_B(x)=$
  $1-(1- \mu_A(x))\ (1- \mu_B(x))$
- $\mu_{A \cap B}(x)=\mu_A(x)\ \mu_B(x)$
- Για $\mu_A(x)$, $\mu_B(x)$ μεταξύ 0 και 1, $\mu_{A \cup B}(x)$ και $\mu_{A \cap B}(x)$ επίσης στο διάστημα

Fuzzy Information Retrieval

- Fuzzy sets are modeled based on a thesaurus
- This thesaurus is built as follows:
  - Let c(i,l) be a normalized correlation factor for (k_i ,k_l):

$$c(i,l) = \frac{n(i,l)}{n_i + n_l - n(i,l)}$$

    - n_i: number of docs which contain k_i
    - n_l: number of docs which contain k_l
    - n(i,l): number of docs which contain both k_i and k_l
- We now have the notion of *proximity* among index terms.

Fuzzy Information Retrieval

- The correlation factor c(i,l) can be used to define fuzzy set membership for a document dj as follows:

$$\mu_i(j) = 1 - \prod_{k_l \in d_j} (1 - c(i,l))$$

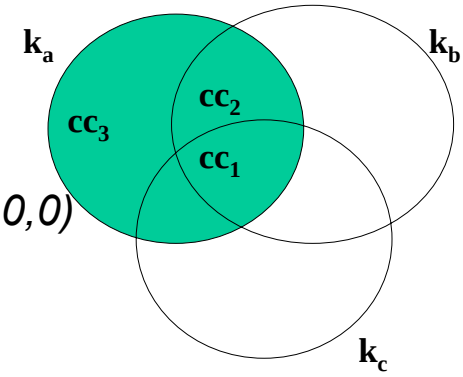
  - $\mu_i(j)$: membership of doc d_j in fuzzy subset associated with k_i
- The above expression computes an algebraic sum over all terms in the doc d_j
- A doc d_j belongs to the fuzzy set for k_i, if its own terms are associated with k_i

Fuzzy Information Retrieval

- $$\mu_i(j) = 1 - \prod_{k_l \in d_j} (1 - c(i,l))$$

  $\mu_i(j)$: membership of doc d_j in fuzzy subset associated with k_i
- If doc d_j contains a term k_l which is closely related to k_i, we have
  - $c(i,l) \sim 1$
  - $\mu_i(j) \sim 1$
  - index k_i is a good fuzzy index for doc

Fuzzy IR: An Example



- $q = k_a \wedge (k_b \vee \neg k_c)$
- $qdnf = (1,1,1) \vee (1,1,0) \vee (1,0,0)$
  $= cc_1 \vee cc_2 \vee cc_3$
- $\mu_q(dj) = \mu_{cc_1 \vee cc_2 \vee cc_3}(dj) =$

$1- (1-\mu_{cc_1}(dj)) (1-\mu_{cc_2}(dj)) (1-\mu_{cc_3}(dj)) =$

$1 - (1 - \mu_a(dj)\ \mu_b(dj)\ \mu_c(dj)) * (1 - \mu_a(dj)\ \mu_b(dj)\ (1- \mu_c(dj))$

$*(1 - \mu_a(dj)\ (1- \mu_b(dj))\ (1- \mu_c(dj))$

Fuzzy Information Retrieval

- Fuzzy IR models have been discussed mainly in the literature associated with fuzzy theory
- Experiments with standard test collections are not available
- Difficult to compare at this time

Extended Boolean Model

- Boolean model is simple and elegant.
- But, no provision for a ranking
- As with the fuzzy model, a ranking can be obtained by relaxing the condition on set membership
- Extend the Boolean model with the notions of partial matching and term weighting
- Combine characteristics of the Vector model with properties of Boolean algebra
  - interpret conjunctions and disjunctions in terms of Euclidean distances

Extended Boolean Model

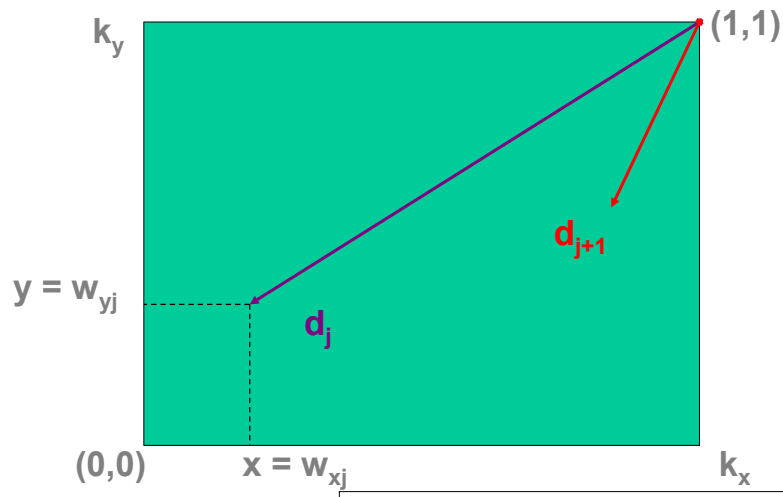
- The *extended Boolean model* is based on a critique of a basic assumption in Boolean algebra. Let:

$q = k_x \wedge k_y$

$w_{xj} = f_{xj} * \frac{idf(x)}{max(idf(i))}$ associated with $[k_x, d_j]$

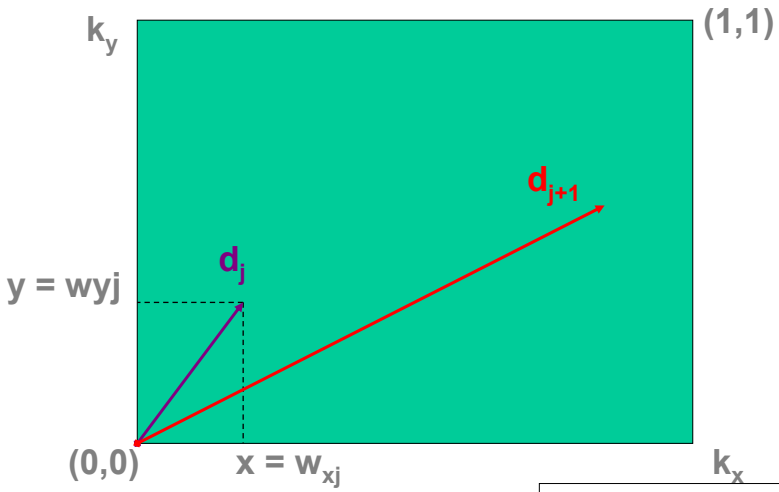
To lighten up the notation: $w_{xj} = x$ and $w_{yj} = y$

Extended Boolean Model



$$sim(q_{and},d_j) = 1 - sqrt(\frac{(1-x)^2 + (1-y)^2}{2})$$

Extended Boolean Model



$$sim(q_{or},d_j) = sqrt(\frac{x^2 + y^2}{2})$$

Conclusions

- Model is quite powerful
- Properties are interesting and might be useful
- Computation is somewhat complex
- However, distributivity operation does not hold for ranking computation:
  - $q_1 = (k_1 \vee k_2) \wedge k_3$
  - $q_2 = (k_1 \wedge k_3) \vee (k_2 \wedge k_3)$
  - $sim(q_1,d_j) \neq sim(q_2,d_j)$

ΠΙΘΑΝΟΤΙΚΟ ΜΟΝΤΕΛΟ

- $D=\{d_1, d_2, d_3, \ldots, d_n\}$ το σύνολο των κειμένων της συλλογής.
- $K=\{k_1, k_2, \ldots, k_t\}$ το σύνολο των λέξεων κλειδιών
- w_{iq} και w_{ij}: δυαδικά βάρη (0 ή 1)
- Κάθε ερώτηση q ορίζει ένα σύνολο R σχετικών κειμένων
- Διαλέγουμε τυχαία ένα κείμενο από τη συλλογή D:
  - $P(d_j)$: η πιθανότητα να έχουμε επιλέξει το κείμενο d_j
  - P(R): η πιθανότητα το κείμενο που επιλέξαμε να ανήκει στο σύνολο R των σχετικών κειμένων
  - $P(R \mid d_j)$: δεδομένου ότι επιλέξαμε το κείμενο d_j, ποια η πιθανότητα να είναι και σχετικό ή πιο απλά **ποια η πιθανότητα το dj να είναι σχετικό με το ερώτημα**
  - $P(d_j \mid R)$: δεδομένου ότι επιλέξαμε ένα κείμενο σχετικό, ποια η πιθανότητα το κείμενο αυτό να είναι το dj
  - $P(k_i \mid R)$: δεδομένου ότι επιλέξαμε ένα κείμενο σχετικό, ποια η πιθανότητα η λέξη κλειδί ki να είναι μεταξύ των λέξεων κλειδιών του επιλεγμένου κειμένου

- Στο πιθανοτικό μοντέλο:
  - *sim(q,d_j) = P(d_j σχετικό με q) / P(d_j μη σχετικό με q) δηλ.*
  - *sim(q,d_j) = P(R |d_j) / P(¬R |d_j) όπου* ¬R είναι το συμπλήρωμα του R
  - Θεώρημα Bayes:

    P(R |d_j) =P(R ∩ d_j)/P(d_j) και

    P(d_j |R) =P(R ∩ d_j)/P(R) άρα

    P(R|d_j) = (P(d_j |R) P(R))/ P(d_j) . Επομένως

    sim(q,d_j) = P(d_j |R) P(R)/ P(d_j | ¬R)P(¬R) ~

    P(d_j |R) / P(d_j | ¬R) αφού P(R) και P(¬R) εξαρτώνται μόνο από το q και όχι από το κείμενο d_j
  - *Ο τελεστής ~ : είναι ανάλογο του*

- Υποθέτουμε δεν υπάρχουν συσχετίσεις/εξαρτήσεις μεταξύ των λέξεων κλειδιών δηλ. ανεξάρτητες λέξεις - κλειδιά.

$$sim(q,dj) \sim \frac{P(dj \mid R)}{P(dj \mid \neg R)}$$

$$= \prod_{i=1,w_{ij}=1}^{t} \frac{P(k_i \mid R)}{P(k_i \mid \neg R)} \prod_{i=1,w_{ij}=0}^{t} \frac{P(\neg k_i \mid R)}{P(\neg k_i \mid \neg R)} \sim$$

$$\prod_{i=1,w_{ij}=1}^{t} \frac{P(k_i \mid R)}{P(k_i \mid \neg R)} \prod_{i=1,w_{ij}=0}^{t} \frac{P(\neg k_i \mid R)}{P(\neg k_i \mid \neg R)} \prod_{i=1}^{t} \frac{P(\neg k_i \mid \neg R)}{P(\neg k_i \mid R)}$$

$$\sim \prod_{i=1,w_{ij}=1}^{t} \frac{P(k_i \mid R)P(\neg k_i \mid \neg R)}{P(k_i \mid \neg R)P(\neg k_i \mid R)} \quad (\text{Υποθέτουμε } P(k_i \mid R) = P(k_i \mid \neg R), \text{ if } w_{iq} = 0)$$

$$\approx \prod_{i=1,w_{iq}=1,w_{ij}=1}^{t} \frac{P(k_i \mid R)P(\neg k_i \mid \neg R)}{P(k_i \mid \neg R)P(\neg k_i \mid R)}$$

$$= \prod_{i=1,w_{iq}=1,w_{ij}=1}^{t} \frac{P(k_i \mid R)(1-P(k_i \mid \neg R))}{P(k_i \mid \neg R)(1-P(k_i \mid R))}$$

- Λαμβάνοντας τον λογάριθμο της προηγούμενης παράστασης έχουμε:

$$sim(q,dj) \sim \sum_{i=1}^{t} w_{ij} w_{iq} \left(\log \frac{P(k_i \mid R)}{1-P(k_i \mid R)} + \log \frac{1-P(k_i \mid \neg R)}{P(k_i \mid \neg R)} \right)$$

Αρχική Κατάταξη

- Πιθανότητες $P(k_i \mid R)$ και $P(k_i \mid \neg R)$;
- Υποθέτουμε αρχικά:
  - $P(k_i \mid R) = 0.5$
  - $P(k_i \mid \neg R) = \frac{n_i}{N}$

    όπου n_i ο αριθμός των κειμένων που περιέχουν τη λέξη κλειδί k_i
  - Με βάση αυτή την υπόθεση, ανάκτηση και κατάταξη των σχετικών κειμένων
  - Επόμενα βήματα: Βελτίωση της εκτίμησης των παραπάνω πιθανοτήτων άρα και της σχετικότητας των ανακτώμενων κειμένων

Βελτίωση της αρχικής κατάταξης

- Αν
  - V : το σύνολο των κειμένων που κρίθηκαν ως σχετικά
  - V_i : το υποσύνολο αυτών των κειμένων που περιέχουν τη λέξη κλειδί k_i
- Επανεκτίμηση των πιθανοτήτων:
  - $P(k_i \mid R) = \frac{V_i}{V}$
  - $P(k_i \mid \neg R) = \frac{n_i - V_i}{N - V}$
- Συνεχή επανάληψη μέχρι τη σταθεροποίηση των αποτελεσμάτων

Πλεονεκτήματα Μειονεκτήματα

- Πλεονεκτήματα:
  - Τα κείμενα κατατάσσονται κατά φθίνουσα σειρά ως προς την πιθανότητα σχετικότητας
- Μειονεκτήματα:
  - Απαραίτητη η αρχική εκτίμηση των $P(k_i \mid R)$
  - Δεν λαμβάνονται υπόψη οι όροι tf and idf.