Matching Functions

LIS466 - Information Retrieval

Week 5

Models, Classification & Clustering

Models, Classification & Clustering

- Prefatory to today
  - You've read about IR models; query and document collection representations

Models, Classification & Clustering

- Prefatory to today
  - You've read about IR models; query and document collection representations
- Today
  - Matching functions
  - Project update

Models, Classification & Clustering

- Prefatory to today
  - You've read about IR models; query and document collection representations
- Today
  - Matching functions
  - Project update
- Next Time
  - Ranking documents - Blended

Retrieval Models: Boolean

Retrieval Models: Boolean

- *Aka Exact Match - simplest form of IR*

Retrieval Models: Boolean

- *Aka Exact Match - simplest form of IR*
- *True/False* AND OR NOT

Retrieval Models: Boolean

- *Aka Exact Match - simplest form of IR*
- *True/False* AND OR NOT
- lincoln

Retrieval Models: Boolean

- *Aka Exact Match - simplest form of IR*
- *True/False* AND OR NOT
- lincoln
  - president AND lincoln

Retrieval Models: Boolean

- *Aka Exact Match - simplest form of IR*
- *True/False* AND OR NOT
- lincoln
  - president AND lincoln
  - Ford Motor Company announces new Lincoln Mercury

Retrieval Models: Boolean

- *Aka Exact Match - simplest form of IR*
- *True/False* AND OR NOT
- lincoln
  - president AND lincoln
  - Ford Motor Company announces new Lincoln Mercury
  - president AND lincoln AND NOT (automobile OR car)

Example

- *Let's experiment using different online search engines*

Vector Space Model

- Started in the 1960s, 70; simple, intuitive, less common now
- A document D is represented by a number of index terms
  - Di = (di1, di2, ... din)
  - Create term/document matrix

$$Cos(D_i Q) = \frac{\sum_{j=1}^{t} d_{ij} \bullet q_j}{\sqrt{\sum_{j=1}^{t} d_{ij}^{\,2} \cdot \sum_{j=1}^{t} q_j^{\,2}}}$$

Vector Space Model - weighting

$$tf_{ik} = \frac{f_{ik}}{\sum_{j=1}^{t} f_{ik}}$$

$idfk = \log N/nk$

Vector Space Model - weighting

- Many ways - usually tf•idf

$$tf_{ik} = \frac{f_{ik}}{\sum_{j=1}^{t} f_{ik}}$$

$$idfk = \log N/nk$$

Vector Space Model - weighting

- Many ways - usually tf•idf
- The normalized count of the term occurrences in a document ... where tfik is the term frequency weight of term k in document Di, and fik is the number of occurrences of term k in the document

$$tf_{ik} = \frac{f_{ik}}{\sum_{j=1}^{t} f_{ik}}$$

$$idf_k = \log N/n_k$$

VM term weighting

$$d_{ik} = \frac{(\log(f_{ik}) + 1) \cdot \log(N / n_k)}{\sqrt{\sum_{k=1}^{t} [(\log(f_{ik}) + 1.0) \cdot \log(N / n_k)]^2}}$$

Often queries are modified

Often queries are modified

- Implicit assumption that terms in query are in the document and that they're good indicators of relevance

Often queries are modified

- Implicit assumption that terms in query are in the document and that they're good indicators of relevance
- Modify queries (relevance feedback) creates a kind of baseline of the optimal query, the average vector representing the relevant documents and average vector of non-relevant documents.
  - Rocchio algorithm, 1971

Probabilistic Models

Probabilistic Models

- Theoretical drifting to empirical-based models

Probabilistic Models

- *Theoretical drifting to empirical-based models*
- *Robertson, 1977/97:* Probability Ranking Principle

Probabilistic Models

- *Theoretical drifting to empirical-based models*
- *Robertson, 1977/97:* Probability Ranking Principle
- *Assumption that documents have* some *degree, or probability, of association to queries.*

Probabilistic Models

- *Theoretical drifting to empirical-based models*
- *Robertson, 1977/97:* Probability Ranking Principle
- *Assumption that documents have* some *degree, or probability, of association to queries.*
- *Often treated as a classification issue: classify something as relevant or as not relevant ...*

Probabilistic Models

- *Theoretical drifting to empirical-based models*
- *Robertson, 1977/97:* Probability Ranking Principle
- *Assumption that documents have* some *degree, or probability, of association to queries.*
- *Often treated as a classification issue: classify something as relevant or as not relevant ...*
- *And then the conditions under which it's relevant....*

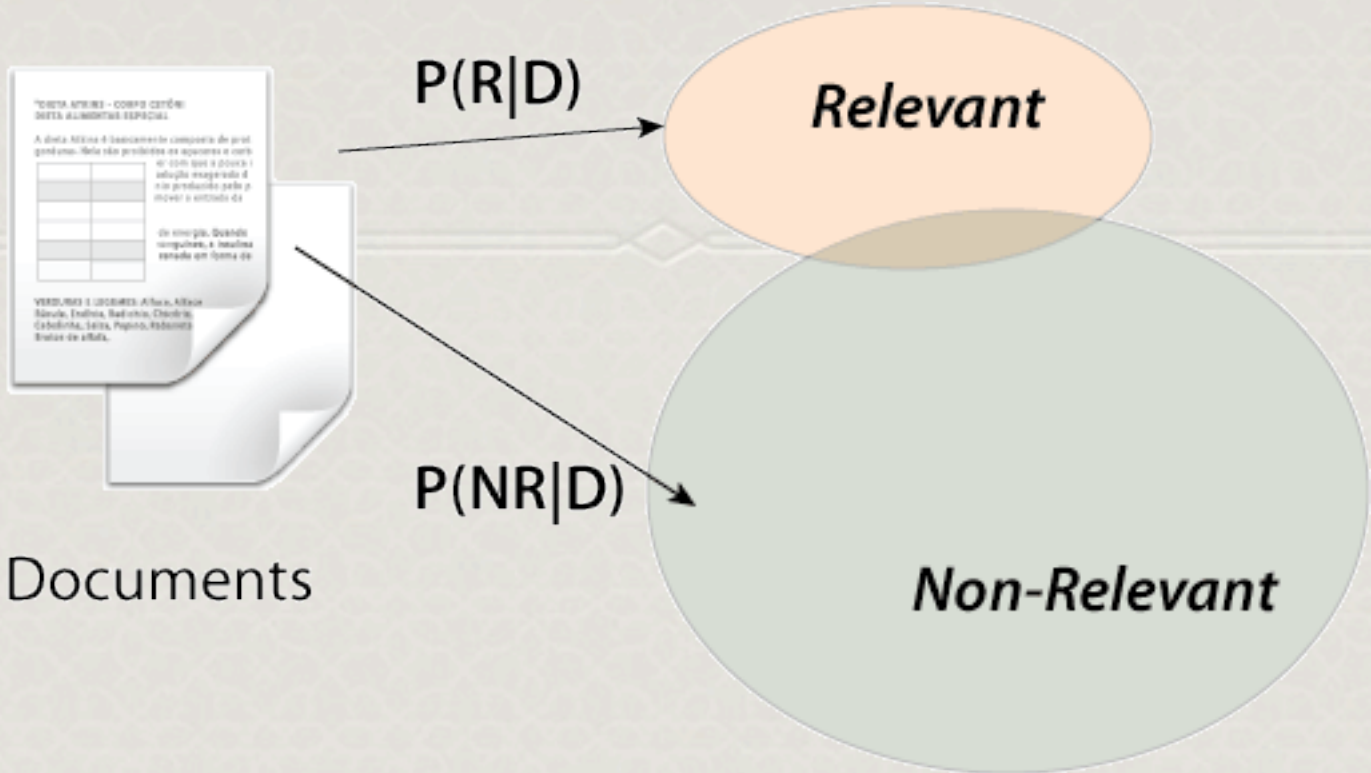
Probabilistic Models

Probabilistic Models

- *A document D is relevant if* P(R | D) > P(NR | D), *where* P(R|D) *is a conditional probability representing the probability of relevance given the representation of that document, and* P(NR|D) *is the conditional probability of non-relevance.*

Probabilistic Models

- *A document D is relevant if* P(R | D) > P(NR | D), *where* P(R|D) *is a conditional probability representing the probability of relevance given the representation of that document, and* P(NR|D) *is the conditional probability of non-relevance.*
- *Aka Bayes Decision Rules (or here a Bayes classifier)*



So, how to calculate the relevant documents?

"Likelihood ratio"

- P(R|D) = P(D|R)P(R) / P(D)
  - *(Skipping all the other formula & variants)*
- *Contingency Table*

	Relevant	Non-Rel	Total
di=1	ri	ni - ri	ni
di=0	R-ri	N-ni-R+ri	N-ri
Total	R	N-R	N

BM25

- TREC - set of retrieval experiments
- BM25 - used in commercial search engines

$$\sum_{i \in Q} \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \bullet \frac{(k_i + 1) f_i}{K + f_i} \cdot \frac{(k_2 + 1) qf_i}{k_2 + qf_i}$$

"Language Models"

"Language Models"

- *Text from a variety of "language technologies" such as speech recognition, machine translation, and handwriting.*

"Language Models"

- *Text from a variety of "language technologies" such as speech recognition, machine translation, and handwriting.*
- *Simplest form known as* unigram *language model - the probability distribution over the words in the language*

"Language Models"

- *Text from a variety of "language technologies" such as speech recognition, machine translation, and handwriting.*
- *Simplest form known as* unigram *language model - the probability distribution over the words in the language*
- *Associates a probability of occurrence with every word in the index vocabulary*

"Language Models"

- *Text from a variety of "language technologies" such as speech recognition, machine translation, and handwriting.*
- *Simplest form known as* unigram *language model - the probability distribution over the words in the language*
- *Associates a probability of occurrence with every word in the index vocabulary*
- *Sum of probabilities of terms (cat + food vs. cat + girl)*

"Language Models"

- *Text from a variety of "language technologies" such as speech recognition, machine translation, and handwriting.*
- *Simplest form known as* unigram *language model - the probability distribution over the words in the language*
- *Associates a probability of occurrence with every word in the index vocabulary*
- *Sum of probabilities of terms (cat + food vs. cat + girl)*
- *bi-gram (predicting based on previous term; tri-gram...)*

Query Likelihood Ranking

- *Rank documents by the probability that the query text could be generated by the document language model ... In other words, we calculate the probability that we could pull the query words out of the "bucket" of words representing the document.*

Relevance Models & Pseudo-Relevance Feedback

- *Emphasis on the representation of* the topic.
- *Using Relevance to affect the probability ranking ...*
  - P(w|R)logP(w|D)
  - *P = probability*
  - w|R = *weight given some Relevance input*
  - w|D = *weight given some Document*

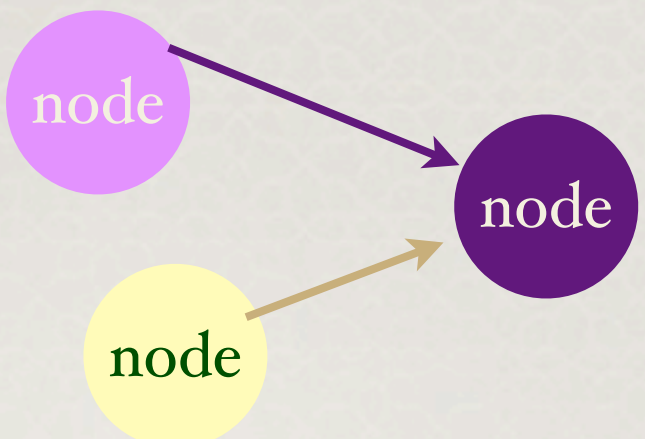
Evidence ...

Evidence ...

- *Some kind of evidence to affect rankings - such as* inference
- (Looking at the cat around feeding time, someone says "I just bought --- ---.") Starts to say "cat." We infer "food" as the next term.

Evidence ...

- *Some kind of evidence to affect rankings - such as* inference
- (Looking at the cat around feeding time, someone says "I just bought --- ---.") Starts to say "cat." We infer "food" as the next term.



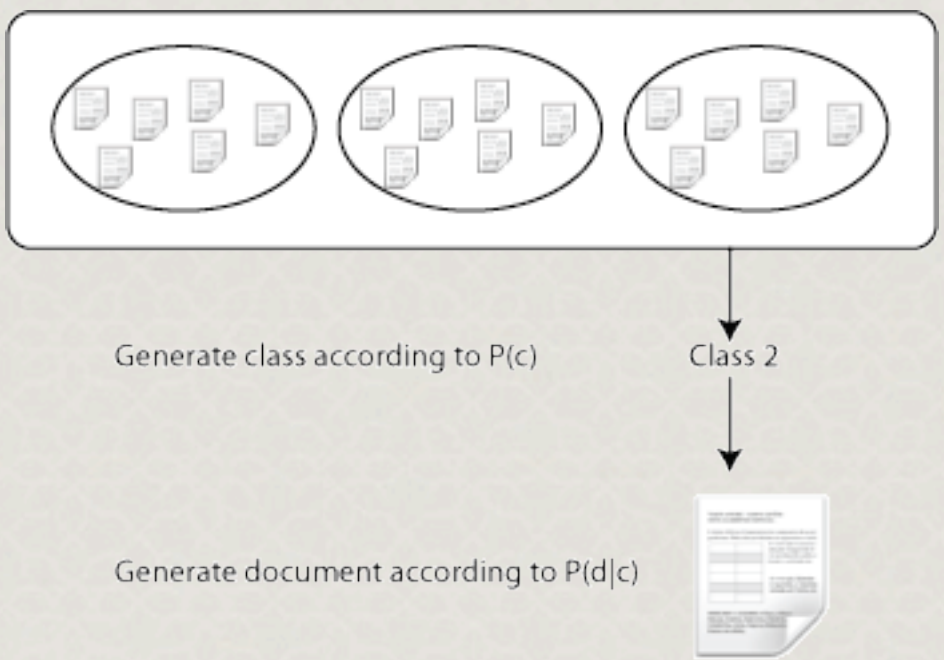
Inference network with 3 nodes

Humans or Machine?!

- Machine Learning
- Train the software to classify new documents based on evidence from a training set.
  - Two classes (those that are + relevant and those that are - not)
  - Part of data mining
  - Visualization, too.
  - Be aware of "overtraining" the system.

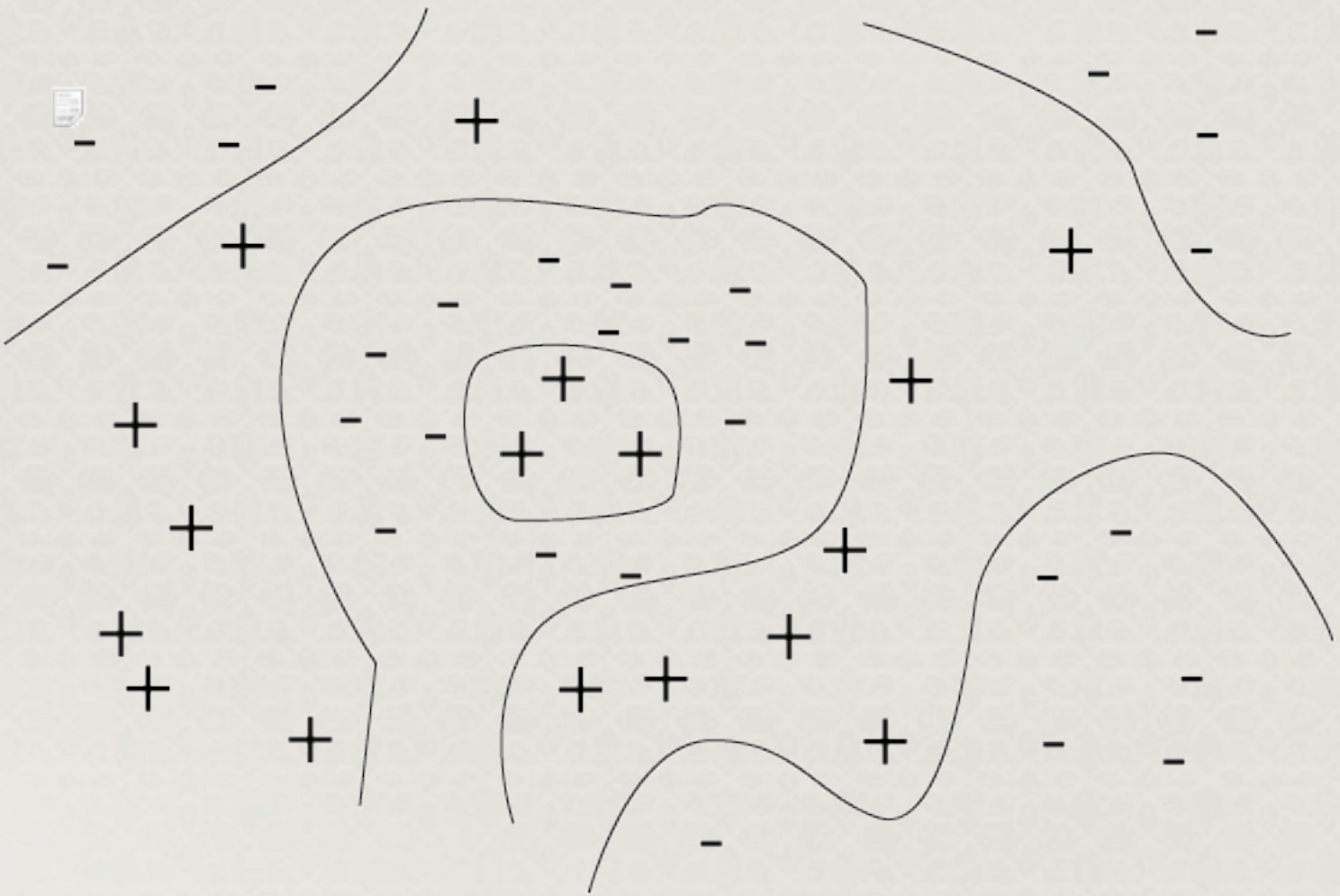
Clustering

- *Too many to mention but here are some examples.*



Nearest neighbor clustering

+ = relevant



Questions? And some suggestions

Questions? And some suggestions

- *Read the suggested texts, of course, and look at earlier version of this course for a very long list of articles.*

Questions? And some suggestions

- *Read the suggested texts, of course, and look at earlier version of this course for a very long list of articles.*
- *See Sparck Jones's* Simple, proven techniques

Questions? And some suggestions

- *Read the suggested texts, of course, and look at earlier version of this course for a very long list of articles.*
- *See Sparck Jones's* Simple, proven techniques
- *See also new monographs about Interactive Visualization or texts that emphasize the union of data and the interface*

Questions? And some suggestions

- *Read the suggested texts, of course, and look at earlier version of this course for a very long list of articles.*
- *See Sparck Jones's* Simple, proven techniques
- *See also new monographs about Interactive Visualization or texts that emphasize the union of data and the interface*
- *Work through examples: e.g., compare searching across languages - the language, technical, user, and query behaviors*

Questions? And some suggestions

- *Read the suggested texts, of course, and look at earlier version of this course for a very long list of articles.*
- *See Sparck Jones's* Simple, proven techniques
- *See also new monographs about Interactive Visualization or texts that emphasize the union of data and the interface*
- *Work through examples: e.g., compare searching across languages - the language, technical, user, and query behaviors*
- SIGIR, D-Lib, *other journals*

Project - Marc example

Project - Marc example

- Let's take sections from MARC records and add weights to 'em

Project - Marc example

- Let's take sections from MARC records and add weights to 'em
- What parts of a MARC record do you think ...

Project - Marc example

- Let's take sections from MARC records and add weights to 'em
- What parts of a MARC record do you think ...
  - Represent the document itself

Project - Marc example

- Let's take sections from MARC records and add weights to 'em
- What parts of a MARC record do you think ...
  - Represent the document itself
  - Represent the intellectual content

Project - Marc example

- Let's take sections from MARC records and add weights to 'em
- What parts of a MARC record do you think ...
  - Represent the document itself
  - Represent the intellectual content
  - Represent another representation (such as "concepts") from the work

Project - Marc example

- Let's take sections from MARC records and add weights to 'em
- What parts of a MARC record do you think ...
  - Represent the document itself
  - Represent the intellectual content
  - Represent another representation (such as "concepts") from the work
  - Would be useful for searching?

Project - Marc example

- Let's take sections from MARC records and add weights to 'em
- What parts of a MARC record do you think ...
  - Represent the document itself
  - Represent the intellectual content
  - Represent another representation (such as "concepts") from the work
  - Would be useful for searching?
  - Other features or values?

Project - MARC example

- *Follow this link to see more about MARC - what parts should we extract?*
- *http://www.loc.gov/marc/bibliographic/examples.html*
- http://www.loc.gov/marc/bibliographic/examples.html