Today’s topics

• Generalized query operators
  – Evidence accumulation and structured queries
• Basics of Bayesian networks
  – Bayesian nets for Text Retrieval
• Structured+unstructured queries
  – Adding database-like queries
User has an information need

Translates need to query

Translation depends on IR system’s syntax + user’s sophistication

System presents docs meeting query

Key: system’s model of query-doc proximity

Navigate

Update query

Browse

Review
Models of query-doc proximity

- **Boolean queries**
  - Doc is either in or out for query

- **Vector spaces**
  - Doc has non-negative proximity to query

- **Evidence accumulation**
  - Combine score from multiple sources

- **Bayesian nets and probabilistic methods**
  - Infer probability that doc meets user’s information need
Evidence accumulation

- View each term in the query as providing partial evidence of match
- $tf \times idf +$ vector space retrieval is one example
  - Corpus-dependent ($idf$ depends on corpus)
- In some situations corpus-dependent evidence is undesirable
Corpus-independent evidence

- When is corpus-independent scoring useful?
  - When corpus statistics are hard to maintain
    - Distributed indices - more later
    - Rapidly changing corpora
  - When stable scores are desired
    - Users get used to issuing a search and seeing a doc with a score of (say) 0.9303
    - User subsequently filters by score
      - “Show me only docs with score at least 0.9”
Corpus-independent scoring

• Document routing is a key application
  – There is a list of standing queries
    • e.g., *bounced check* in a bank’s email customer service department
  – Each incoming doc (email) scored against all standing queries
  – Routed to destination (customer specialist) based on best-scoring standing query

• More on this with automatic classification
Typical corpus-independent score

• Use a convex function of $tf_{ij}$
  – e.g., $\text{Score}(i,j) = 1 - \exp(-a \times tf_{ij})$
  – $a$ is a tuning constant
  – gives a contribution of query term $i$ for doc $j$

• Given a multi-term query, compute the average contribution, over all query terms
Bayesian Networks for Text Retrieval

• Text retrieval
  – Find the best set of documents that satisfies a user’s information need

• Bayesian Network
  – Model causal relationship between events
  – Infer the belief that an event holds based on observations of other events
What is a Bayesian network?

• Is a directed acyclic graph
• Nodes
  – Events or Variables
    • Assume values.
    • For our purposes, all Boolean
• Links
  – model dependencies between nodes
Toy Example

\[
\begin{array}{c c}
 f & 0.3 \\
 \neg f & 0.7 \\
\end{array}
\]

![Diagram of a toy example with nodes for Finals, Project Due, No Sleep, Gloom, and Triple Latte, and probabilities labeled on edges and nodes.](image-url)
Links as dependencies

• Link Matrix
  – Attached to each node
    • Give influences of parents on that node.
  – Nodes with no parent get a “prior probability”
    • e.g., f, d.
  – interior node: conditional probability of all combinations of values of its parents
    • e.g., n, g, t.
Independence Assumption

- Variables not connected by a link: no direct conditioning.
- Joint probability - obtained from link matrices.
- See examples on next slide.
Independence Assumption

- Independence assumption: \( P(t|g f) = P(t|g) \)
- Joint probability
  \( P(f d n g t) = P(f) P(d) P(n|f) P(g|f d) P(t|g) \)
Chained inference

- Evidence - a node takes on some value
- Inference
  - Compute belief (probabilities) of other nodes
    - conditioned on the known evidence
  - Two kinds of inference: **Diagnostic** and **Predictive**
- Computational complexity
  - General network: NP-hard
  \[\Rightarrow\] polytree networks - tractable.
Bayes’ theorem: for any two events $a, c$

$$P(a \mid c) = P(a \mid c)P(c) = P(c \mid a)P(a)$$

Implies, for instance:

$$P(a \mid c) = \frac{P(c \mid a)P(a)}{P(c)}$$
Diagnostic Inference

• Propagate beliefs through parents of a node
• Inference rule

\[ P(a) \sum_{b_i} P(c \mid b_i)P(b_i \mid a) \]

\[ P(a \mid c) = \frac{P(c \mid b_i)P(b_i \mid a)}{P(c)} \]
Evidence: $n=\text{true}$
Belief: $P(f|n)=?\)
Diagnostic inference

Inference Rule

\[ P(f | n) = \frac{P(f)}{P(n)} \cdot \frac{P(n | f)}{P(n)} = \frac{0.27}{P(n)} \]

\[ P(\neg f | n) = \frac{P(\neg f)}{P(n)} \cdot \frac{P(n | \neg f)}{P(n)} = \frac{0.21}{P(n)} \]

Normalize

\[ P(f | n) + P(\neg f | n) = 1 \]

\[ \Rightarrow P(n) = 0.48 \]

Beliefs

\[ P(f | n) = 0.56 \]

\[ P(\neg f | n) = 0.44 \]
Predictive Inference

- Compute belief of child nodes of evidence
- Inference rule

\[ P(c \mid a) = \sum_{b} P(c \mid b_i)P(b_i \mid a) \]
Model for Text Retrieval

• Goal
  – Given a user’s information need (evidence), find probability a doc satisfies need

• Retrieval model
  – Model docs in a document network
  – Model information need in a query network
Bayesian nets for text retrieval

Documents

- $d_1$
- $d_2$

Terms

- $r_1$
- $r_2$
- $r_3$

Concepts

- $c_1$
- $c_2$
- $c_3$

Query operators

- $q_1$
- $q_2$

Information need

$\text{AND/OR/NOT}$
Link matrices and probabilities

- Prior doc probability
  \[ P(d) = \frac{1}{n} \]

- \( P(r|d) \)
  - within-document term frequency
  - \( tf \times idf \) - based

- \( P(c|r) \)
  - 1-to-1
  - thesaurus

- \( P(q|c) \): canonical forms of query operators
Example

Document Network

Hamlet
- reason
- reason

Macbeth
- trouble
- double
- two

Query Network

User query
- OR
- NOT
Extensions

• Prior probs don’t have to be $1/n$.
• “User information need” doesn’t have to be a query - can be words typed, in docs read, any combination …
• Link matrices can be modified over time.
  – User feedback.
• The promise of “personalization”
Computational details

• Document network built at indexing time
• Query network built/scored at query time
• Representation:
  – Link matrices from docs to any single term are like the postings entry for that term.
Exercise

• Consider ranking docs for a 1-term query. What is the difference between
  – A cosine-based vector-space ranking where each doc has $tf \times idf$ components, normalized;
  – A Bayesian net in which the link matrices on the docs-to-term links are normalized $tf \times idf$?
Semi-structured search

- Structured search - search by restricting on attribute values, as in databases.
- Unstructured search - search in unstructured files, as in text.
- Semi-structured search: combine both.
Terminology

• Each document has
  – structured *fields* (aka *attributes, columns*)
  – free-form text

• Each field assumes one of several possible *values*
  – e.g., language (French, Japanese, etc.); price (for products); date; …

• Fields can be *ordered* (price, speed), or *unordered* (language, color).
Queries

- A query is any combination of
  - text query
  - field query
- A field query specifies one or more values for one or more fields
  - for numerical values, ranges possible
    - e.g., \textit{price} < 5000.
Example

- Find all docs in corpus with
  - \textit{Price} < 10000
  - \textit{Year} > 1996
  - \textit{Model} = Toyota, and
  - text matches (\textit{excellent OR good NEAR condition}).

- Don’t want to hit underlying database.
  - Demo.
Indexing: structured portion

- For each fields, order docs by values for that field
  - e.g., sorted by authors’ names, language …
- Maintain range indices (in memory) for each value of each attribute
  - like a postings entry
  - counts are like \( freq \) in postings.
Query processing

• Given value for each field, determine counts of matching docs
• Process query using optimization heuristics
  – Lightest axis first
• Merge with text search postings.
Numerical attributes

• Expensive to maintain a separate postings for each value of a numerical attribute
  – e.g., price
• Bucket into numerical ranges, maintain postings for each bucket
• At the user interface, present only bucket boundaries
  – e.g., if index buckets price into steps of $5000, present only these buckets to user
General ranges

• If the UI allows the user to specify an arbitrary numerical range
  – in the used-car section of cars.com: price, year
  – e.g., price between 1234 and 5678.

• Need to walk through the postings entry for (say) the bucket 0-5000, until 1234 reached

• At most two postings entries need a walk-through
Resources

- MIR 2.6, 2.8.
Bayesian Resources


[http://www.aaai.org/Library/Magazine/Vol12/12-04/vol12-04.html](http://www.aaai.org/Library/Magazine/Vol12/12-04/vol12-04.html)


D. Heckerman. A Tutorial on Learning with Bayesian Networks.

Microsoft Technical Report MSR-TR-95-06

