Probabilistic Information Retrieval
Part I: Survey

Alexander Dekhtyar
department of Computer Science
University of Maryland

Outline

- ✔**Part I: Survey**:
  - – Why use probabilities ?
  - – Where to use probabilities ?
  - – How to use probabilities ?
- ✔**Part II: In Depth**:
  - – Probability Ranking Principle
  - – Boolean Independence Retrieval model

Why Use Probabilities ?

Standard IR techniques

- ✔ Empirical for most part
  - – success measured by experimental results
  - – few properties provable
- ✔ This is not unexpected
- ✔ Sometimes want properties of methods

Probabilistic IR

- ✔ Probabilistic Ranking Principle
  - – provable "minimization of risk"
- ✔ Probabilistic Inference
  - – "justify" your decision
- ✔ Nice theory

Why use probabilities ?

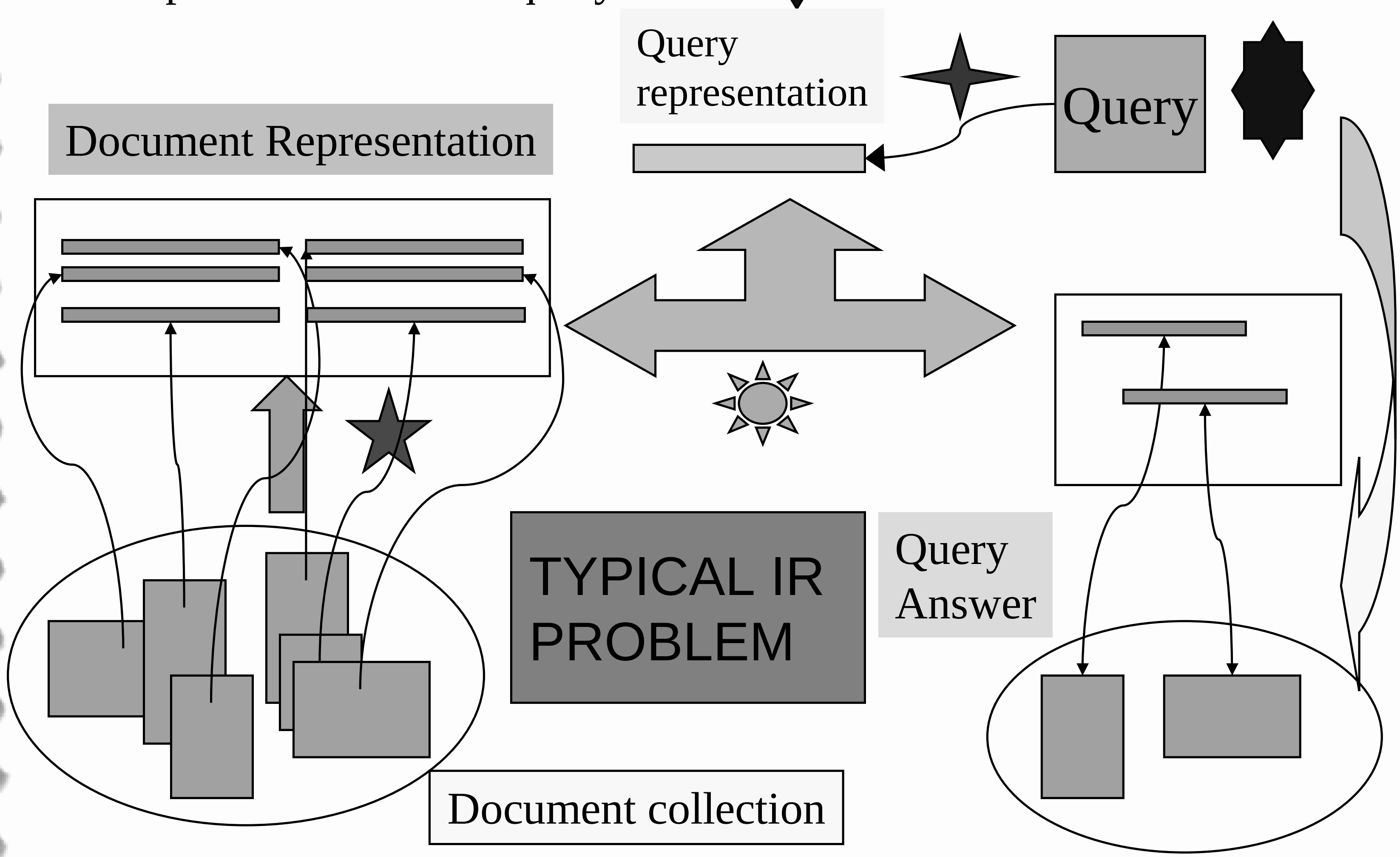
- ✔ Information Retrieval deals with Uncertain Information

How exact is the representation of the document ?

How exact is the representation of the query ?

How well is query matched to data?

How relevant is the result to the query ?



Why use probabilities ?

- ✔ Information Retrieval deals with Uncertain Information
- ✔ Probability theory seems to be the most natural way to quantify uncertainty

try explaining to non-mathematician
what the fuzzy measure of 0.75 means

Probabilistic Approaches to IR

- ✔ Probability Ranking Principle (Robertson, 70ies; Maron, Kuhns, 1959)
- ✔ Information Retrieval as Probabilistic Inference (van Rijsbergen & co, since 70ies)
- ✔ Probabilistic Indexing (Fuhr & Co.,late 80ies-90ies)
- ✔ Bayesian Nets in IR (Turtle, Croft, 90ies)
- ✔ Probabilistic Logic Programming in IR (Fuhr & co, 90ies)

Success : varied

Next: Probability Ranking Principle

Probability Ranking Principle

- ✔ Collection of Documents
- ✔ User issues a query
- ✔ A Set of documents needs to be returned
- ✔ **Question: In what order to present documents to user ?**

Probability Ranking Principle

- ✔ **Question: In what order to present documents to user ?**
- ✔ Intuitively, want the “best” document to be first, second best - second, etc…
- ✔ Need a formal way to judge the “goodness” of documents w.r.t. queries.
- ✔ **Idea: Probability of relevance of the document w.r.t. query**

Probability Ranking Principle

If a reference retrieval system's response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user who submitted the request ...

Probability Ranking Principle

… where the probabilities are estimated as accurately a possible on the basis of whatever data made available to the system for this purpose ...

Probability Ranking Principle

… then the **overall effectiveness** of the system to its users **will be the best** that is obtainable on the basis of that data.

W.S. Cooper

Probability Ranking Principle

If a reference retrieval system's **response to each request** is *a ranking of the documents in the collections* in ***order of decreasing probability of usefulness*** to the user who submitted the request ...

… where the probabilities are estimated as accurately a possible on the basis of whatever data made available to the system for this purpose ...

… then the **overall effectiveness** of the system to its users **will be the best** that is obtainable on the basis of that data.
W.S. Cooper

Probability Ranking Principle

How do we do this ?

???????????????????

Let us remember Probability Theory

Let *a*, *b* be two events.

Bayesian formulas

$$p(a \mid b)\, p(b) = p(a \cap b) = p(b \mid a)\, p(a)$$

$$p(a \mid b) = \frac{p(b \mid a)\, p(a)}{p(b)}$$

$$p(\overline{a} \mid b)\, p(b) = p(b \mid \overline{a})\, p(\overline{a})$$

Probability Ranking Principle

Let x be a document in the collection.
Let R represent **relevance** of a document w.r.t. given (fixed) query and let NR represent **non-relevance.**

Need to find p($R|x$) - probability that a retrieved document x is **relevant.**

$$p(R\,|\,x) = \frac{p(x\,|\,R)\,p(R)}{p(x)}$$

$$p(NR\,|\,x) = \frac{p(x\,|\,NR)\,p(NR)}{p(x)}$$

p(R),p(NR) - prior probability of retrieving a (non) relevant document

p($x|R$), $p(x|NR)$ - probability that if a relevant (non-relevant) document is retrieved, it is x.

Probability Ranking Principle

$$p(R \mid x) = \frac{p(x \mid R)\, p(R)}{p(x)}$$

$$p(NR \mid x) = \frac{p(x \mid NR)\, p(NR)}{p(x)}$$

Ranking Principle (Bayes' Decision Rule):

If p(*R*|*x*) > p(*NR*|*x*) then *x* <u>is relevant</u>, otherwise *x* <u>is not relevant</u>

Probability Ranking Principle

<u>Claim:</u> ***PRP minimizes the average probability of error***

$$p(error \mid x) = \begin{cases} p(R \mid x) & \text{If we decide } \boldsymbol{NR} \\ p(NR \mid x) & \text{If we decide } \boldsymbol{R} \end{cases}$$

$$p(error) = \sum_{x} p(error \mid x)\, p(x)$$

p(error) is minimal when all *p(error|x)* are minimimal.
<u>Bayes' decision rule</u> minimizes each *p(error|x)*.

PRP: Issues (Problems?)

- ✔ How do we compute all those probabilities?
  - Cannot compute exact probabilities, have to use estimates.
  - **Binary Independence Retrieval (BIR)** (to be discussed in **Part II**)
- ✔ Restrictive assumptions
  - “Relevance” of each document is independent of relevance of other documents.
  - Most applications are for Boolean model.
  - “Beatable” (Cooper’s counterexample, is it well-defined?).

Next: Probabilistic Indexing

Probabilistic Indexing

- ✔ Probabilistic Retrieval:
  - – ***Many Documents - One Query***
- ✔ Probabilistic Indexing:
  - – ***One Document - Many Queries***
- ✔ *Binary Independence Indexing* (BII):dual to *Binary Independence Retrieval* (**part II**)
- ✔ *Darmstadt Indexing (DIA)*
- ✔ *n-Poisson Indexing*

Next: Probabilistic Inference

Probabilistic Inference

- ✔ Represent each document as a collection of sentences (formulas) in some logic.
- ✔ Represent each query as a sentence in the same logic.
- ✔ Treat Information Retrieval as a **process of inference**: document D is relevant for query Q if $p(D \rightarrow Q)$ is high in the inference system of selected logic.

Probabilistic Inference: Notes

- ✔ $p(D \rightarrow Q)$ is the probability that the description of the document in the logic implies the description of the query.
  - $\rightarrow$ is not material implication:

$$p(A \rightarrow B) = \frac{p(A \wedge B)}{p(A)} \neq p(\neg A \vee B)$$

- ✔ Reasoning to be done in some kind of probabilistic logic.

Probabilistic Inference: Roadmap

- ✔ Describe your own probabilistic logic/inference system
  - – document / query representation
  - – inference rules
- ✔ Given query Q compute $p(D \rightarrow Q)$ for each document D
- ✔ Select the "winners"

Probabilistic Inference:Pros/Cons

Pros:

- ✔ Flexible: Create-Your-Own-Logic approach
- ✔ Possibility for provable properties for PI based IR.
- ✔ Another look at the same problem ?

Cons:

- ✔ Vague: PI is just a broad framework not a cookbook
- ✔ Efficiency:
  - Computing probabilities always hard;
  - Probabilistic Logics are notoriously inefficient (up to being undecidable)

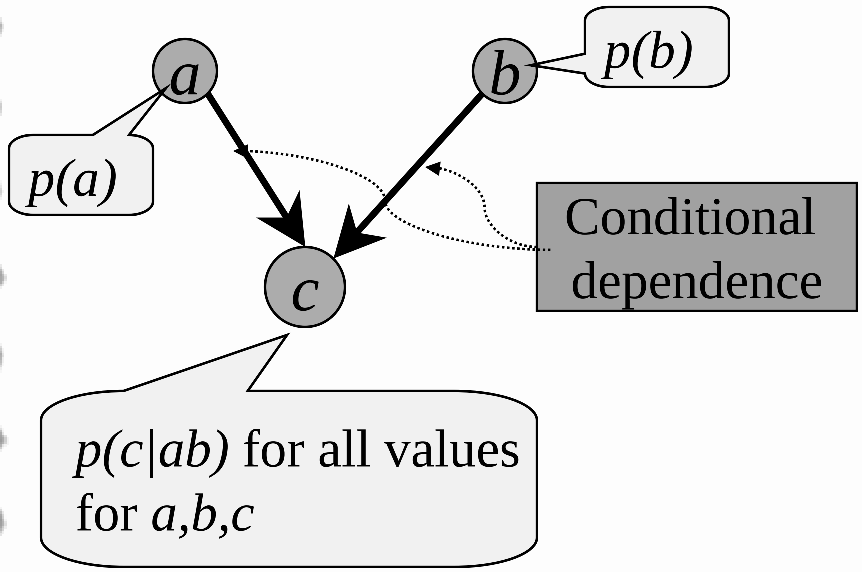
Next: Bayesean Nets In IR

Bayesian Nets in IR

- ✔ Bayesian Nets is the most popular way of doing probabilistic inference in AI.
- ✔ What is a Bayesian Net ?
- ✔ How to use Bayesian Nets in IR?

Bayesian Nets

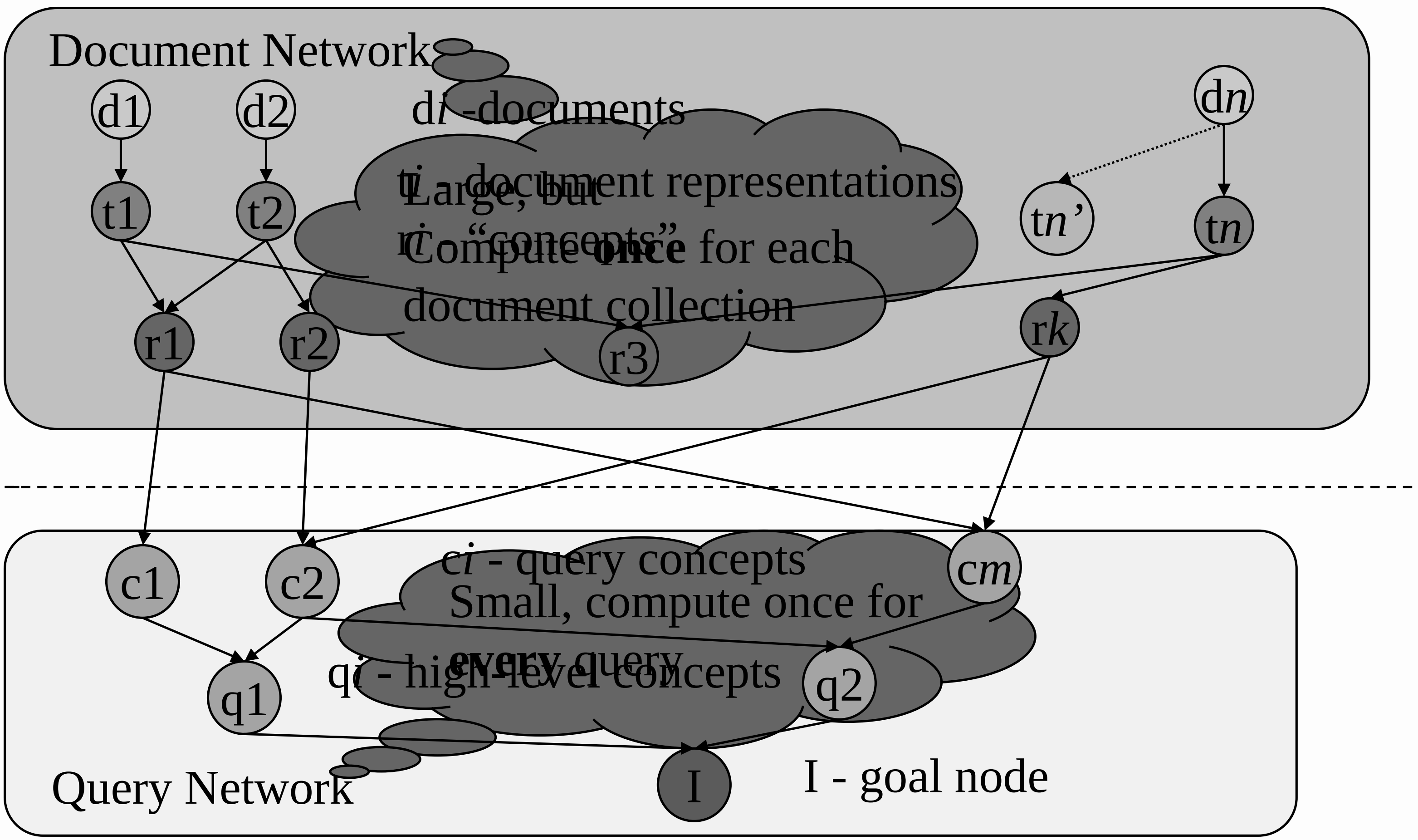
a,*b*,*c* - propositions (events).



- Running Bayesian Nets:
  - Given probability distributions for roots and conditional probabilities can compute apriori *probability* of any instance
  - Fixing assumptions (e.g., *b* was observed) will cause recomputation of probabilities

For more information see J. Pearl, "***Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference***", 1988, Morgan-Kaufman.

Bayesian Nets for IR: Idea



Bayesian Nets for IR: Roadmap

- ✔ Construct Document Network (once !)
- ✔ For each query
  - – Construct best Query Network
  - – Attach it to Document Network
  - – Find subset of **d*i*’s** which maximizes the probability value of node **I** (best subset).
  - – Retrieve these **d*i*’s** as the answer to query.

Bayesian Nets in IR: Pros / Cons

•**Pros**

- ✔ More of a cookbook solution
- ✔ Flexible:create-your- own Document (Query) Networks
- ✔ Relatively easy to update
- ✔ Generalizes other Probabilistic approaches
  - – PRP
  - – Probabilistic Indexing

•**Cons**

- ✔ Best-Subset computation is NP-hard
  - – have to use quick approximations
  - – approximated Best Subsets may not contain best documents
- ✔ Where Do we get the numbers ?

Next: Probabilistic Logic Programming in IR

Probabilistic LP in IR

- ✔ Probabilistic Inference estimates $p(D \rightarrow Q)$ in some probabilistic logic
- ✔ Most probabilistic logics are hard
- ✔ **Logic Programming:** possible solution
  - – logic programming languages are restricted
  - – but decidable
- ✔ Logic Programs may provide flexibility (write your own IR program)
- ✔ Fuhr & Co: Probabilistic Datalog

Probabilistic Datalog: Example

- Sample Program:

```
0.7 term(d1,ir).
0.8 term(d1,db).
0.5 link(d2,d1).
about(D,T):- term(D,T).
about(D,T):- link(D,D1), about(D1,T).
```

- Query/Answer:

```
:- term(X,ir) & term(X,db).
X= 0.56 d1
```

Probabilistic Datalog: Example

- Sample Program:

```
0.7 term(d1,ir).
0.8 term(d1,db).
0.5 link(d2,d1).
about(D,T):- term(D,T).
about(D,T):- link(D,D1), about(D1,T).
```

- Query/Answer:

```
q(X):- term(X,ir).
q(X):- term(X,db).
:-q(X)
X= 0.94 d1
```

Probabilistic Datalog: Example

- Sample Program:

```
0.7 term(d1,ir).
0.8 term(d1,db).
0.5 link(d2,d1).
about(D,T):- term(D,T).
about(D,T):- link(D,D1), about(D1,T).
```

- Query/Answer:

```
:- about(X,db).
X= 0.8 d1;
X= 0.4 d2
```

Probabilistic Datalog: Example

- Sample Program:

```
0.7 term(d1,ir).
0.8 term(d1,db).
0.5 link(d2,d1).
about(D,T):- term(D,T).
about(D,T):- link(D,D1), about(D1,T).
```

- Query/Answer:

```
:- about(X,db)& about(X,ir).
X= 0.56 d1
X= 0.28 d2  # NOT 0.14 = 0.7*0.5*0.8*0.5
```

Probabilistic Datalog: Issues

- ✔ Possible Worlds Semantics
- ✔ Lots of restrictions (!)
  - – all statements are either independent or disjoint
    - not clear how this is distinguished syntactically
  - – point probabilities
  - – needs to carry a lot of information along to support reasoning because of independence assumption

Next: Conclusions (?)

Conclusions (Thoughts aloud)

- ✔ IR deals with uncertain information in many respects
- ✔ Would be nice to use probabilistic methods
- ✔ Two categories of Probabilistic Approaches:
  - Ranking/Indexing
    - Ranking of documents
    - No need to compute exact probabilities
    - Only estimates
  - Inference
    - logic- and logic programming-based frameworks
    - Bayesian Nets
- ✔ Are these methods useful (and how)?

Next: Survey of Surveys

Probabilistic IR: Survey of Surveys

- ✔ Fuhr (1992) ***Probabilistic Models In IR***
  - – BIR, PRP, Indexing, Inference, Bayesian Nets, Learning
  - – Easier to read than most other surveys.
- ✔ Van Rijsbergen, chapter 6 of IR book: ***Probabilistic Retrieval***
  - – PRP, BIR, Dependence treatment
  - – most math
  - – no references past 1980 (1977)
- ✔ Crestani,Lalmas,van Rijsbergen, Campbell, (1999) ***Is this document relevant?... Probably"…***
  - – BIR, PRP, Indexing, Inference, Bayesian Nets, Learning
  - – Seems to repeat Fuhr and classic works word-by-word

Probabilistic IR: Survey of Surveys

<u>General Problem with probabilistic IR surveys:</u>

- ✔ Only “old” material rehashed;
- ✔ No “current developments”
  - – e.g. logic programming efforts not surveyed
- ✔ Especially true of the last survey