Intelligent Agents Probabilistic Information Retrieval

Prof. Dr. Ralf Möller Universität zu Lübeck Institut für Informationssysteme



Agents





Classical planning



Given – S, G and set of actions $\{a_i\} =>$ Agent's Model M^R Find – sequence of actions or plan $\pi = \langle a_1, a_2, ..., a_n \rangle$ that transforms S to G.





Joint planning



Given – S, G and set of actions $\{a_i\} =>$ Agent's Model M^R Find – sequence of actions or **joint plan** $\pi = \langle a_1, a_2, ..., a_n \rangle$ that transforms S to G^+ .





Agent model for human behavior anticipation





Agent model for human exceptation anticipation





Human specifies goal: Solve a certain problem



- *M^H* human model of the problem to be solved
- M^{R}_{h} is the human's understanding of the robot's M^{R}
- *M^R* robot model of the problem to be solved
- \tilde{M}_{r}^{H} is the robot's understanding of M^{H} (anticipate human behavior)
- M^{R}_{h} is the robot's understanding of M^{R}_{h} (anticipate human's expectations) INVERSITAT ZU LUBECK INSTITUTE FOR INFORMATIONSSYSTEME

What causes differences in mental models?

- Expectations on capabilities / actions
 - Human may have misconceptions about robot's actions
 - Certain actions in human's mental model may not be feasible for the robot
- Expected state of the world
 - Human may assume certain facts are true in the world (even if they are not true)
- Expected goals
 - Human may have misconceptions about the robot's goals/intentions
- Sensor model differences
 - Human may have partial observability of the robot's activities
 - Human may have incorrect beliefs about robot's observational capabilities



Where do mental models come from?

- In certain applications mental models are known beforehand
- Learning simple models for generating explanations/explicability
 - This will be covered later
- Learning full models (transition functions, rewards)
 - Through interaction with the users



Goal Specification

- For specific problem classes M^H , goal specification languages must be developed to specify M^R
 - If goal specifications cannot sensibly be provided by humans for a certain application domain,
 Al researchers do need to continue their work
- We will consider ...
 - how information retrieval (IR) goals can be represented and communicated to a web-mining agent (IR agent)
 - how uncertainty about IR goals can be reduced
 - how reinforcement feedback can be collected by the IR agent



Recap: Document and query representation

Only represent occurrences of terms with incidence matrix?

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Use word counts?

Use word frequencies?

What about terms that occur in certain documents but seldomly in the corpus?



 f_{ij} = relative number of terms t_i in document d_j

 $TF_{ij} = f_{ij} / number of terms in d_j$

 n_i = Number of documents with term i N = Total number of documents

$$IDF_i = \log \frac{N}{n_i}$$

TF.IDF measure

$$w_{ij} = TF_{ij} \cdot IDF_i$$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	13.1	11.4	0.0	0.0	0.0	0.0
Brutus	3.0	8.3	0.0	1.0	0.0	0.0
Caesar	2.3	2.3	0.0	0.5	0.3	0.3
Calpurnia	0.0	11.2	0.0	0.0	0.0	0.0
Cleopatra	17.7	0.0	0.0	0.0	0.0	0.0
mercy	0.5	0.0	0.7	0.9	0.9	0.3
worser	1.2	0.0	0.6	0.6	0.6	0.0





Why probabilities in IR?



In traditional IR systems, matching between each document and query is attempted in a semantically imprecise space of index terms

Probabilities provide a principled foundation for uncertain reasoning *Can we use probabilities to quantify our uncertainties?*



Probability Ranking Principle

- Collection of Documents
- User issues a query
- A set of documents is found and needs to be returned
- Question: In what order to present documents to user ?
- Need a formal way to judge the "goodness" of documents w.r.t. queries
- Idea: Probability of relevance of the documents w.r.t. query



Ben He, Probability Ranking Principle, Reference Work Entry, Encyclopedia of Database Systems, Ling Liu, Tamer, Öszu (Eds.), 2168-2169, Springer, **2009**.

Probabilistic Approaches to IR

 Probability Ranking Principle (Robertson, 70ies; Maron, Kuhns, 1959)

Robertson S.E. The probability ranking principle in IR. J. Doc., 33:294–304, **1977**.

M. E. Maron and J. L. Kuhns. On Relevance, Probabilistic Indexing and Information Retrieval. *J. ACM* 7, 3, 216-244, **1960**.

• IR as Probabilistic Inference

(van Rijsbergen & et al., since 70ies)

van Rijsbergen C.J. Inform. Retr.. Butterworths, London, 2nd edn., **1979**.

• Probabilistic IR (Croft, Harper, 70ies)

Croft W.B. and Harper D.J. Using probabilistic models of document retrieval without relevance information. J. Doc., 35:285–295, **1979**.

• Probabilistic Indexing (Fuhr & et al., late 80ies-90ies)

Norbert Fuhr. 1989. Models for retrieval with probabilistic indexing. *Inf. Process. Manage*. 25, 1, 55-72, **1989**.



Let us recap probability theory

• Bayesian probability formulas

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

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

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

• Odds:

$$O(y) = \frac{p(y)}{p(\overline{y})} = \frac{p(y)}{1 - p(y)}$$



Odds vs. Probabilities





Let x be a document in the retrieved collection. Let R represent Relevance=true of a document w.r.t. given (fixed) query and let NR represent Relevance=false.

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

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

p(*R*),p(*NR*) - prior probability of retrieving a relevant or nonrelevant document, respectively

p(x|R), p(x|NR) - probability that if a relevant or 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 is relevant, If $p(R|x) \le p(NR|x)$ then x is not relevant

• Note:
$$p(R | x) + p(NR | x) = 1$$



Claim: PRP minimizes the average probability of error $p(error \mid x) = \begin{cases} p(R \mid x) & \text{If we decide NR} \\ p(NR \mid x) & \text{If we decide R} \end{cases}$

Expected overall error

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

p(error) is minimal when all p(error|x) are minimal Bayes' decision rule minimizes each p(error|x).

> Ranking Principle (Bayes' Decision Rule): If p(R|x) > p(NR|x) then x is relevant, If $p(R|x) \le p(NR|x)$ then x is not relevant

Probability Ranking Principle

- More complex case: retrieval costs
 - C: cost of retrieval of <u>relevant</u> document
 - C': cost of retrieval of <u>non-relevant</u> document
 - *d*: a document
- Documents *d* are ranked according the to the Probability Ranking Principle when it holds that :
 - If $C \cdot p(R \mid d) + C' \cdot (1 p(R \mid d)) \le C \cdot p(R \mid d') + C' \cdot (1 p(R \mid d'))$ for any other d' not yet retrieved,

then

d is the next document to be retrieved



Relevance models

- Given: **PRP** to be applied
 - "Relevance" of each document is independent of relevance of other documents
- Need to estimate probability: P(R|q,d)
- Binary Independence Retrieval (BIR):
 - Many documents D one query q
 - Estimate P(R|q,d) by considering whether $d \in D$ is relevant for q
- Binary Independence Indexing (BII):
 - One document d many queries Q
 - Estimate P(R|q,d) by considering whether a document d is relevant for a query $q \in Q$



- "Binary" = Boolean: documents are represented as binary vectors of terms:
 - $\vec{x} = (x_1, \dots, x_n)$
 - $x_i = 1$ iff term *i* is present in document *x*.
- "Independence": terms occur in documents independently
- Different documents can be modeled as same vector.



- Queries: binary vectors of terms
- Given query q,
 - for each document d need to compute p(Relevant=true|q,d)
 - replace with computing p(Relevant=true|q,x) where x is vector representing d
- Interested only in ranking
- Will use odds (the higher, the better):

$$O(R \mid q, \vec{x}) = \frac{p(R \mid q, \vec{x})}{p(NR \mid q, \vec{x})} = \frac{p(R \mid q)}{p(NR \mid q)} \cdot \frac{p(\vec{x} \mid R, q)}{p(\vec{x} \mid NR, q)}$$



$$O(R \mid q, \vec{x}) = \frac{p(R \mid q, \vec{x})}{p(NR \mid q, \vec{x})} = \begin{bmatrix} p(R \mid q) \\ p(NR \mid q) \\ p(NR \mid q) \end{bmatrix} \begin{bmatrix} p(\vec{x} \mid R, q) \\ p(\vec{x} \mid NR, q) \end{bmatrix}$$

Constant for each query

• Using Independence Assumption:

$$\frac{p(\vec{x} \mid R, q)}{p(\vec{x} \mid NR, q)} = \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$$

So: $O(R \mid q, d) = O(R \mid q) \cdot \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$



$$O(R | q, d) = O(R | q) \cdot \prod_{i=1}^{n} \frac{p(x_i | R, q)}{p(x_i | NR, q)}$$

• Since *x_i* is either 0 or 1:

$$O(R \mid q, d) = O(R \mid q) \cdot \prod_{x_i=1} \frac{p(x_i = 1 \mid R, q)}{p(x_i = 1 \mid NR, q)} \cdot \prod_{x_i=0} \frac{p(x_i = 0 \mid R, q)}{p(x_i = 0 \mid NR, q)}$$

• Let $p_i = p(x_i = 1 \mid R, q); \quad r_i = p(x_i = 1 \mid NR, q);$

• Assume, for all terms not occurring in the query ($q_i=0$) $p_i = r_i$







All matching terms







• Optimize Retrieval Status Value (RSV):

$$RSV = \log \prod_{x_i = q_i = 1} \frac{p_i(1 - r_i)}{r_i(1 - p_i)} = \sum_{x_i = q_i = 1} \log \frac{p_i(1 - r_i)}{r_i(1 - p_i)}$$

• All boils down to computing RSV.

$$RSV = \log \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} = \sum_{x_i=q_i=1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$
$$RSV = \sum_{x_i=q_i=1} C_i; \quad \text{For all query terms i:} \\ \text{Find docs containing term i } (\rightarrow \text{ inverted index}) \\ \text{Non-standard Databases and Data Mining}$$
$$c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)} = \log \frac{p_i}{(1-p_i)} + \log \frac{(1-r_i)}{r_i}$$

So, how do we compute ci's from our data?



- Estimating RSV coefficients: Groundtruth for subset of docs
 - It is known wether docs are relevant or not
- For each term i look at the following table:

Document	Relevant	Non-Relevant	Total
Xi=1	S	n-s	п
$X_i = \theta$	S-s	N-n-S+s	N-n
Total	S	N-S	N

$$p_i = p(x_i = 1 | R, q); \quad r_i = p(x_i = 1 | NR, q);$$

• Estimates:
$$p_i \approx \frac{s}{S}$$
 $r_i \approx \frac{(n-s)}{(N-S)}$

- Estimating RSV coefficients.
- For each term *i* look at the following table:

Document	Relevant	Non-Relevant	Total
$X_i=1$	S	n-s	п
$X_i = \theta$	S-s	N-n-S+s	N-n
Total	S	N-S	N

• Estimates:
$$p_i \approx \frac{s}{S}$$
 $r_i \approx \frac{(n-s)}{(N-S)}$ $c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$
 $c_i \approx K(N, n, S, s) = \log \frac{s/(S-s)}{(n-s)/(N-n-S+s)}$



Avoid division by 0

$$c_i \approx K(N, n, S, s) = \log \frac{(s+1/2)/(S-s+1/2)}{(n-s+1/2)/(N-n-S+s+1/2)}$$



Estimation in practice $p_i \approx \frac{s}{S}$ $r_i \approx \frac{(n-s)}{(N-S)}$ $c_i = \log \frac{p_i}{(1-p_i)} + \log \frac{(1-r_i)}{r_i}$

 If non-relevant documents are approximated by the whole collection (S=s=0), then r_i (prob. of occurrence term i in non-relevant documents for query) is n/N and

 $- \log (1-r_i)/r_i = \log (N-n)/n \approx \log(1+(N-n)/n) = \log N/n = IDF$

- Idea cannot be easily extended to p_i
- Estimate p_i (probability of occurrence of term i in relevant docs):
 - From relevant documents if we know some
 - Use constant 0.5 then just get idf weighting of terms
 (p_i and 1-p_i cancel out)

$$RSV = \sum_{x_i = q_i = 1} \log \frac{N}{n_i}$$

- ...

ERSITÄT ZU LÜBECK

• We have a nice theoretical foundation of TF.IDF (in the binary case: TF=1 or TF=0)

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. In *Document retrieval systems*, Vol. 3. Taylor Graham Publishing, London, UK, UK 132-142. **1988**. Greiff, Warren R., A Theory of Term Weighting Based on Exploratory Data Analysis. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 11-19, **1998**.

Robertson S.E., Understanding inverse document frequency: On theoretical arguments for idf. J. Doc., 60:503–520, **2004**.

Iteratively estimating *p*_i

Expectation Maximization:

- 1. Assume that p_i constant over all q_i in query
 - $p_i = 0.5$ (even odds) for any given doc
- 2. Determine guess of relevant document set from subset V:
 - V is fixed size set of highest ranked documents on this model
- 3. We need to improve our guesses for p_i and r_i , so
 - Use distribution of q_i in docs in V. Let V_i be set of documents containing q_i
 - $\bullet \quad p_i = \left| V_i \right| \, / \, \left| V \right|$
 - Assume if not retrieved then not relevant
 - $\bullet \quad r_i \ = (n_i \left|V_i\right|) \ / \ (N \left|V\right|)$
- 4. Go to 2. until convergence then return ranking



Probabilistic Relevance Feedback

- 1. Guess a preliminary probabilistic description of *R* and use it to retrieve a first set of documents V, as above.
- 2. Interact with the user to refine the description: learn some definite members of R and NR
- 3. Reestimate p_i and r_i on the basis of these
 - Or can combine new information with original guess (use Bayesian prior):

$$p_i^{(2)} = \frac{|V_i| + \lambda p_i^{(1)}}{|V| + \lambda}$$

λ is prior weight

Repeat, thus generating a succession of approximations to *R*.


Binary Independence Indexing

- "Learning" from queries
 - More queries: better results

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

- p(q|x,R) probability that if document x had been deemed relevant, query q had been asked
- The rest of the framework is similar to BIR



Binary Independence Retrieval vs. Binary Independence Indexing

<u>BIR</u>

- Many Documents, One Query
- Bayesian Probability:

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

- Varies: document representation
- Constant: query (representation)

BII

- One Document,
 Many Queries
- Bayesian Probability



Constant: document



PRP and BIR/BII: The lessons

- Getting reasonable approximations of probabilities is possible.
- Simple methods work only with restrictive assumptions:
 - term independence
 - terms not in query do not affect the outcome
 - boolean representation of documents/queries
 - document relevance values are independent
- Some of these assumptions can be removed



Bayesian Nets in IR

Inference Network Model

Howard Turtle, and W. Bruce Croft. Inference networks for document retrieval. In *Proc. SIGIR*, pp. 1-24. ACM Press. **1989**.

Howard Turtle, and W. Bruce Croft. Evaluation of an inference network-based retrieval model. *TOIS* 9 (3): 187-222. **1991**.

Belief Network Model

Berthier A. N. Ribeiro and Richard Muntz. A belief network model for IR. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '96). ACM, New York, NY, USA, 253-260. **1996**.

Bayesian Network Retrieval Model

Luis M.de Campos, Juan M.Fernández-Luna, Juan F.Huete, Clustering terms in the Bayesian network retrieval model: a new approach with two term-layers, Applied Soft Computing, Volume 4, Issue 2, Pages 149-158, May **2004**.

Some subsequent slides are based on a presentation "An Overview of Bayesian Network-based Retrieval Models" by Juan Manuel Fernández Luna



Inference Network Model



Inference Network Model: "reason trouble -two"

Inference Network Model

- Construct document network (once !)
- For each query
 - Construct query network (on the fly !)
 - Attach it to document network
 - Find doc subset d_is maximizing P(Q | d_is) (best subset)
 - Retrieve these d_is as the answer to query.
- But:
 - Powerset of docs defines huge search space
 - Exact answers for queries P(Q | d_is) rather "expensive"
 - BN structure has loops (no polytree)

Howard Turtle, and W. Bruce Croft. Inference networks for document retrieval. In *Proc. SIGIR*, pp. 1-24. ACM Press. **1989**.

Howard Turtle, and W. Bruce Croft. Evaluation of an inference network-based retrieval model. *TOIS* 9 (3): 187-222. **1991**.

Probability Distributions:

NIVERSITÄT ZU LÜBECK

- <u>Term nodes</u>: $p(t_j)=1/m, p(\neg t_j)=1-p(t_j)$
- Document nodes: $p(D_j | Pa(D_j)), \forall D_j$

 $p(d_j | Q) = p(d_j, Q) / p(Q) = \alpha p(d_j, Q) = \alpha \Sigma_\tau p(d_j, Q, \tau)$

$$= \alpha \Sigma_{\tau} p(d_j | \tau) p(Q | \tau) p(\tau) \qquad \tau = (t_1, \dots, t_m)$$

But... If a document has been indexed by, say, 30 "most important" terms, we need to estimate (and store) 2³⁰ probabilities.

Berthier A. N. Ribeiro and Richard Muntz. A belief network model for IR. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '96). ACM, New York, NY, USA, 253-260. **1996**.

Summary

- Refinement of the agent architecture
- Mental models
- Human specifies goal of the agents
 - Example IR agent Human specifies query
 - Agent's goal: Satisfy information need defined by query
- Foundations of IR
 - Probability ranking principle
- IR with probabilistic models (BNs)
 - Example: Inference network model
 - Example: Belief network model
 - Next: BN Retrieval Model (with recommendations)

Uncertainty about goal: re-estimate information need

Intelligent Agents Probabilistic Information Retrieval

Prof. Dr. Ralf Möller Universität zu Lübeck Institut für Informationssysteme

Bayesian Network Retrieval Model

the Bayesian network retrieval model: a new approach with two term-layers, Applied Soft Computing, Volume 4, Issue 2, Pages 149-158, May **2004**.

Bayesian Network Retrieval Model

Removing the term independency restriction:

 We are interested in representing the main relationships among terms in the collection.

PRP and Recommendations

Those who retrieved d_i were also interested in d_i

Summary: Probabilistic Information Retrieval

- PRP defines a well-defined framework for IR
 - Can understand pragmatic approaches (e.g., TF.IDF)
 - Can be used for formalizing IR (What is the IR problem?)
 - Provides for means to compute ranking of results
- Agents can use different models and different QA strategies for IR
- We will see soon:
 - Agents can update internal models by reinforcement feedback
 - Agents can adapt strategies to new user queries (new goals to be expected)

Recap: Agent architecture $M^{R} = IR(Q)$

Action: Return retrieved docs

Waiting for feedback (possibly with follow-up queries)

Retrieved docs to be evaluated by human

IM FOCUS DAS LEBEN 54

Challenges of Human-Aware Al Systems: Subbarao Kambhampati

Confusion Matrix (e.g., for Classification)

In the example confusion matrix below, of the 8 actual cats, the system predicted that three were dogs, and of the six dogs, it predicted that one was a rabbit and two were cats. We can see from the matrix that the system in question has trouble distinguishing between cats and dogs, but can make the distinction between rabbits and other types of animals pretty well.

Example confusion

matrix

	Cat	Dog	Rabbit
Cat	5	3	0
Dog	2	3	1
Rabbit	0	2	11

Understanding where an agent has deficiencies

(Direct) feedback: Present confusion matrix to an agent

Reinforcement:

Relevance feedback for retrieval results (agent might build confusion matrix internally)

Unranked retrieval evaluation: Precision and Recall

 $recall = \frac{Number of relevant documents retrieved}{Total number of relevant documents}$

 $precision = \frac{Number of relevant documents retrieved}{Total number of documents retrieved}$

Unranked retrieval evaluation: Precision and Recall

- Precision: fraction of retrieved docs that are relevant = P(retr&rel|retrieved)
- Recall: fraction of relevant docs that are retrieved = P(retr&rel|relevant in repos)

	Relevant	Not Relevant
Retrieved	true positives (tp)	false positives (fp)
Not Retrieved	false negatives (fn)	true negatives (tn)

- Precision = tp/(tp + fp)
- Recall = tp/(tp + fn)

Relative operating characteristic (ROC)

- What if goal specification involves control parameters?
- E.g., for strategies
- Investigate effects of parameter adjustments
- Compare TP rate and FP rate
- Example w/ three strategies
- Measure: Area under curve (AUC) curve = ROC

Wikipedia

Overview on evaluation measures

		True condition			
	Total population	Condition positive	Condition negative	$= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	True positive	False positive (Type I error)	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$
	Predicted condition negative	False negative (Type II error)	True negative	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	$\frac{\text{Negative predictive value}}{(\text{NPV})} = \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$
	Accuracy (ACC) =	$\frac{\text{True positive rate (TPR),}}{\text{Sensitivity, Recall}} = \frac{\Sigma \text{True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{TPR}{FPR}$	Diagnostic odds ratio (DOR)
	$\frac{\Sigma \text{ Irue positive } + \Sigma \text{ Irue negative}}{\Sigma \text{ Total population}}$	False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR–) $= \frac{FNR}{TNR}$	$=\frac{LR^{+}}{LR^{-}}$

[Wikipedia]

Precision and Recall

- Determining Recall can be difficult
- Total number of relevant items is sometimes not available use pooling
 - Sample across the database and perform relevance judgment on these items
 - Apply different retrieval algorithms to the same database for the same query. The aggregate of relevant items is taken as the total relevant set

First instruct agent to do IR on known corpus (train agent to use best QA strategy) Then send it out.

Standard Methodology for Measuring Relevance in IR

- To measure relevance effectiveness of ad-hoc IR, we need:
 - 1. A document collection.
 - 2. A suite of information needs, expressible as **queries**.
 - Must be representative of actual user needs.
 - Sample from query logs, if available.
 - **3. Binary assessments** of either <u>Relevant</u> or <u>Nonrelevant</u> for each query and each document.
 - Can be more nuanced, e.g., 0, 1, 2, 3, ...
 - Use *pooling*, when it is unfeasible to assess every (q, d) pair.

The TREC Benchmark

- TREC: Text REtrieval Conference (<u>http://trec.nist.gov/</u>)
 - Became an annual conference in 1992, co-sponsored by the National Institute of Standards and Technology (NIST) and DARPA.
 - Participants are given parts of a standard set of documents and TOPICS (from which queries have to be derived) in different stages for training and testing.
 - Participants submit the P/R values for the final document and query corpus and present their results at the conference.

Trade-off between Recall and Precision

IM FOCUS DAS LEBEN

Information Retrieval and Web Search, IR Evaluation and IR Standard Text Collections, Rada Mihalcea

F-measure

- One measure of performance that takes into account both recall and precision.
- Harmonic mean of recall and precision:

$$F = \frac{2PR}{P+R} = \frac{2}{\frac{1}{R} + \frac{1}{P}}$$

• Compared to arithmetic mean, both need to be high for harmonic mean to be high.

Ranked Retrieval Measures

- Binary relevance:
 - 11-point Interpolated Precision-Recall Curve
 - R-precision
 - Precision@K (P@K) and Recall@K (R@K)
 - Mean Average Precision (MAP)

Recall-Precision Curves: An Example

IM FOCUS DAS LEBEN

Information Retrieval and Web Search, IR Evaluation and IR Standard Text Collections, Rada Mihalcea

Interpolating a Recall/Precision Curve

- Interpolate a precision value for each *standard recall level*:
 - $r_j \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$

$$- r_0 = 0.0, r_1 = 0.1, \dots, r_{10} = 1.0$$

 The interpolated precision at the *j*-th standard recall level is the maximum known precision at any recall level between the *j*-th and (*j* + 1)-th level:

$$P(r_j) = \max_{r_j \le r \le r_{j+1}} P(r)$$

Interpolating a Recall/Precision Curve: An Example

IM FOCUS DAS LEBEN

Information Retrieval and Web Search, IR Evaluation and IR Standard Text Collections, Rada Mihalcea

Average Recall/Precision Curve

- Typically average performance over a large set of queries.
- Compute average precision at each standard recall level across all queries.
- Plot average precision/recall curves to evaluate overall system performance on a document/query corpus.
- Average:
 - Micro-average: compute P/R/F
 once for the entire set of queries
 - Macro-average: average of within-query precision/recall

How To Compare Two or More Systems

• The curve closest to the upper right-hand corner of the graph indicates the best performance

IM FOCUS DAS LEBEN

Information Retrieval and Web Search, IR Evaluation and IR Standard Text Collections, Rada Mihalcea

R-precision

 Precision at the R-th position in the ranking of results for a query that has R relevant documents.

n	doc #	relevant	
1	588	Х	
2	589	Х	
3	576		
4	590	Х	
5	986		
6	592	Х	
7	984		
8	988		
9	578		
10	985		
11	103		
12	591		
13	772	Х	
14	990		

R = # of relevant docs = 6

R-Precision = 4/6 = 0.67

Precision@K

- 1. Set a rank threshold K.
- 2. Compute % of documents relevant in top K.
 - Ignores documents ranked lower than K.
- Example:
 - Prec@3 of 2/3
 - Prec@4 of 2/4
 - Prec@5 of 3/5

• In a similar way we have Recall@K

Mean Average Precision (MAP)

- 1. Consider rank position of each of the R relevant docs:
 - K₁, K₂, ... K_R
- 2. Compute Precision@K for each $K_1, K_2, \ldots K_R$.
- 3. Average precision = average of P@K.

• MAP is Average Precision across multiple queries.

Average Precision

Ranking #1 = (1.0+0.67+0.75+0.8+0.83+0.6) /6 = 0.78 Ranking #2 = (0.5+0.4+0.5+0.57+0.56+0.6) /6 = 0.5

Mean Average Precision (MAP)

Average precision query 1 = (1.0+0.67+0.5+0.44+0.5)/5 = 0.62Average precision query 2 = (0.5+0.4+0.43)/3 = 0.44MAP = (0.62 + 0.44)/2 = 0.53

Mean Average Precision (MAP)

- If a relevant document never gets retrieved, we assume the precision corresponding to that relevant document to be zero.
- MAP is macro-averaging: each query counts equally.
- A commonly used measure in current IR research, along with P/R/F



Back to IR: Language Models

- A new approach to probabilistic IR, derived from work in automatic speech recognition, OCR and MT
- Statistically model the use of language in a collection to estimate the probability that a query was *generated* from a particular document
- If the query could have come from the document, then that document is likely to be relevant

Acknowledgment: Slides taken from a presentation on "Principles of Information Retrieval" by Ray Larson



Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '98). ACM, New York, NY, USA, 275-281. **1998**.

IM FOCUS DAS LEBEN

• For the original Ponte and Croft Language Models the goal is to estimate:

$$p(Q|M_d)$$

 That is, the probability of a query given the language model of document d. One approach would be to use:

$$p_{ml}(t \mid M_d) = tf_{t,d}$$

• Maximum likelihood estimate of the probability of term t in document d, where $tf_{(t,d)}$ is the term frequency of t in doc d



- The ranking formula then could be:
 - For each document d in the collection...
- There are problems with this (not least of which is that it is zero for any document that doesn't contain all query terms)
- A robust estimator might be the mean probability of *t* in documents containing it (*df_t* is the document frequency of *t*)

$$p(Q|M_d) = \prod_{t \in Q} p_{ml}(t|M_d)$$

$$p_{avg}(t) = \frac{\left(\sum_{d_{(t \in d)}} p_{ml}(t \mid M_d)\right)}{df_t}$$



- There are still problems with this estimator, in that it treats each document with t as if it came from the SAME language model
- The final form with a "risk adjustment" for a term t in a document d is as follows...



• Let,

$$\hat{p}(t \mid M_d) = \begin{cases} p_{ml}(t, d)^{(1.0 - \hat{R}_{t,d})} \times p_{avg}(t)^{\hat{R}_{t,d}} & \text{if } tf_{(t,d)} > 0 \\ \frac{cf_t}{cs} & \text{otherwise} \end{cases}$$
• Where

• Where
the risk is
$$\hat{R}_{t,d} = \left(\frac{1.0}{(1.0 + \bar{f}_t)}\right) \times \left(\frac{\bar{f}_t}{(1.0 + \bar{f}_t)}\right)^{o_{t,d} \to \#lerms(d)}$$

 i.e. the geometric distribution, f_t is the mean term frequency of term t in the docs where t occurs and cf_t is the raw term count of t in the collection and cs is the collection size (in #term tokens)

• Then,
$$\hat{p}(Q | M_d) = \prod_{t \in Q} \hat{p}(t | M_d) \times \prod_{t \notin Q} 1.0 - \hat{p}(t | M_d)$$



The geometric distribution gives the probability that the first occurrence of success requires k independent trials, each with success probability p. If the probability of success on each trial is p, then the probability that the kth trial (out of k trials) is the first success is 1.0

$$\Pr(X=k) = (1-p)^{k-1}p$$

$$\begin{array}{c} 1.0 \\ 0.8 \\ \hline 0.8 \\ \hline 0.6 \\ \hline 0.4 \\ 0.2 \\ 0.0 \\ \hline 0.2 \\ 0.0 \\ \hline 0 \\ 2 \\ 4 \\ 6 \\ 8 \\ 10 \\ \end{array}$$

$$1 - 1/(1+x) = (1+x)/(1+x) - 1/(1+x) = x / (1+x)$$



Ponte and Croft IR

- Given: Query Q
- Iterate over all docs d: Rank according to $\hat{p}(Q|M_d)$
- Possibly return only k best docs
- When compared to a fairly standard TF.IDF retrieval on the TREC collection ...
- ... the Ponte and Croft language model provided significantly better performance
 - 5% more relevant documents were retrieved overall,
 - with about a 20% increase in mean average precision
- But: Notion of relevance is lacking



Lavrenko and Croft LM

 Reclaim ideas of the probability of relevance from earlier probabilistic models and includes them into the language modeling framework with its effective estimation techniques



Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '01). ACM, New York, NY, USA, 120-127. **2001**.

BIR vs. Ponte and Croft

• The basic form of the older probabilistic model (Binary independence model) is

$$P(D \mid R) = \prod_{w \in D} P(w \mid R) \times \prod_{w \notin D} (1.0 - P(w \mid N))$$

While the Ponte and Croft Language Model is very similar

$$P(Q \mid M_d) = \prod_{t \in Q} P(t \mid M_d) \times \prod_{t \notin Q} (1.0 - P(t \mid M_d))$$



Lavrenko and Croft LM

- What distinguishes the two is how the individual word (term) probabilities are estimated
- Basically they estimate the probability of observing a word in the relevant set using the probability of co-occurrence between the words and the query adjusted by collection level information

$$P(t_1, \dots, t_n \mid M_d) = \prod_{i=1}^n \left((1 - \lambda) P(t_i \mid C) + \lambda P(t_i \mid D) \right)$$

- Where λ is a parameter derived from a test collection
- Lurking danger of overtraining (word like "the", "of", "and" or misspellings): focus on modeling terms distinguishing the model from the general model of a collection [Zaragoza et al. 03]

Good and Bad News

- Standard Vector Space Model
 - Empirical for the most part; success measured by results
 - Few properties provable
- Probabilistic Models
 - Advantages
 - Based on a firm theoretical foundation
 - But: construction of the BN is engineering work
 - Theoretically justified optimal ranking scheme
 - Disadvantages
 - Binary word-in-doc weights (not using term frequencies)
 - Often: Independence of terms (can be alleviated)
 - Amount of computation required is high
 - Has never worked convincingly better in practice



- Dealing with terms directly in PRP context leads to a rather large computational effort
- Vector space approaches benefit from dimension reduction (and need it indeed)
- We will cover probabilistic dimension reduction soon: Compute "topics"



Back to IR agents

- Agents make decisions about which documents to select and report to the agents' creators
 - Recommend the k top-ranked documents
 - Ranking according to PRP or LM as explained before
- How to evaluate an agent's performance
 - Externally (agent creator's satisfaction)
 - Internally (relevance feedback, reinforcement)



Web Mining Agents

- Mining in complex networks requires the management of
 - Distributed work (problem decomposition)
 - Autonomous work (no central control, proactive agents)
 - Collaboration between agents (solution sharing, "collective intelligence in the small")
 - Feedback and adaptation (learning by reinforcement)

Weiss, G. "A Multiagent Perspective of Parallel and Distributed Machine Learning". *Agents 98*: 226–230, **1998**.

Klusch, M.; Lodi, S.; Moro, G.. "Agent-Based Distributed Data Mining". *LNCS 2586*: 104–122, **2003**.

Cao, Longbing; Gorodetsky, Vladimir; Mitkas, Pericles A. "Agent Mining: The Synergy of Agents and Data Mining,". *IEEE Intelligent Systems*. **24** (3): 64–72, **2009**.

Cao, Longbing; Weiss, Gerhard; Yu, Philip. "A Brief Introduction to Agent Mining". Journal of Autonomous Agents and Multi-Agent Systems. **25**: 419–424, **2012**.



Collaboration: Measure for inter-judge (dis)agreement

- Kappa measure
 - (Dis)Agreement measure among judges
 - Designed for categorical judgments
 - Corrects for chance agreement
- $\kappa = [P(A) P(E)] / [1 P(E)]$
- P(A) proportion of time judges agree (observed)
- P(E) what agreement would be by chance (hypothetical)
- $\kappa = 0$ for chance agreement, 1 for total agreement
- In statistics many other measures are defined



Cohen, Jacob, "A coefficient of agreement for nominal scales". *Educational and Psychological Measurement* **20** (1): 37–46, **1960**

P(A)? P(E)?

Number of docs	Judge 1	Judge 2
300	Relevant	Relevant
70	Nonrelevant	Nonrelevant
20	Relevant	Nonrelevant
10	Nonrelevant	Relevant



Kappa Example

- P(A) = 370/400 = 0.925
- P(nonrelevant) = (10+20+70+70)/800 = 0.2125
- P(relevant) = (10+20+300+300)/800 = 0.7878
- $P(E) = 0.2125^2 + 0.7878^2 = 0.665$
- $\kappa = (0.925 0.665)/(1 0.665) = 0.776$
- $\kappa > 0.8 = \text{good agreement}$
- 0.67 < κ < 0.8 -> "tentative conclusions"
- Depends on purpose of study
- For >2 judges: average pairwise κ s



Relevance Feedback: Rocchio Algorithm

- The Rocchio algorithm incorporates relevance feedback information into the vector space model.
- Want to maximize sim(Q, C_r) sim(Q, C_{nr}) where C_r and C_{nr} denote relevant and non-relevant doc vectors, respectively
- The optimal query vector for separating relevant and nonrelevant documents (with cosine sim.):

$$\vec{Q}_{opt} = \frac{1}{|C_r|} \sum_{\vec{d}_j \in C_r} \vec{d}_j - \frac{1}{N - |C_r|} \sum_{\vec{d}_j \notin C_r} \vec{d}_j$$

 Q_{opt} = optimal query; C_r = set of rel. doc vectors in corpus; N = collection size

• Unrealistic definition:

We don't know relevant documents in corpus



The Theoretically Best Query





Rocchio 1971 Algorithm (SMART System)

• Useful in practice:

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

- q_m = modified query vector; q₀ = original query vector; α,β,γ: weights (hand-chosen or set empirically); D_r = set of known relevant doc vectors; D_{nr} = set of known irrelevant doc vectors
- New query moves toward relevant documents and away from irrelevant documents
- Tradeoff α vs. β/γ : If we have a lot of judged documents, we want a higher β/γ .
- Term weight can go negative
 - Negative term weights are ignored (set to 0)

Wikipedia: Gerard Salton, The **SMART** (System for the Mechanical Analysis and Retrieval of Text or Salton's Magic Automatic Retriever of Text) Information Retrieval System is an information retrieval system, developed at Cornell University in the **1960s**. Many important concepts in information retrieval were developed as part of research on the SMART system, including the vector space model, relevance feedback, and Rocchio algorithm.



Salton, G. (Ed.). *The SMART retrieval system: Experiments in automatic document processing*. Englewood Cliffs, NJ: Prentice-Hall. 1971.

Relevance feedback on initial query





Positive feedback is more valuable than negative feedback (so, set $\gamma < \beta$; e.g. $\gamma = 0.25$, $\beta = 0.75$).

Many systems only allow positive feedback ($\gamma=0$).



Relevance feedback in vector spaces

- We can modify the query based on relevance feedback and apply standard vector space model.
- Use only the docs that were marked.
- Relevance feedback can improve recall and precision

Any idea for relevance feedback in probabilistic IR?



IR Agents: Summary

- Goal: Fulfill information need of human user
 - Information need specified in various ways (e.g., query vector)
 - Agent employs strategies to best fulfill its goal(s)
- Agent receives reinforcement feedback ("reward") (e.g., as relevance feedback)
- Agent changes its goal fulfillment strategies for dealing with the same or similar goals
 - E.g., by applying the Rocchio Algorithm
- Agent possibly extends its model of the user
- Agent could refine goals to meet expectations
 - Reduce uncertainty
- Agent could contact other agents to acquire new information

