Probability Spaces of Information Retrieval

Gianni Amati

Glasgow, 27 August 2002

Let us agree on what the space of events is in IR

- Sample space and its points (*an experiment*).

- We assign probabilities to the points of the space according to the outcomes of the experiment.

- Probabilities are spread from the sample points to arbitrary events A, which consist of a certain number of sample points.

  – An event in IR is the occurrence or not of certain terms in a piece of text. A collection D of documents can be seen as

  – *a set of experiments* or *a single experiment*.

The event space

- Let T be the set of terms of the language: this is the set of all sample points.

- Let D be a collection of douments containing L tokens (occurrences of terms). Let $o_1, \ldots, o_L$ be the outcomes of our experiments according D.

- Let us consider the event: the term t occurs in an arbitrary outcome o_j.

- The outcomes can be thought as independent trials as we were tossing a coin or extracting balls from an urn.

Bernoulli's model of IR

- $o_1, \ldots, o_L$. Among these L tokens, TF are occurrences of the term t. Its frequency is $f = \frac{TF}{L}$.

- What is the probability p of having t? We can compute the a posteriori probability p of observing the frequency f in the experiment with the Bayes theorem, by maximising the *likelihood*:

$$P(p|f) = B(TF, L, p) = \binom{L}{TF} p^{TF}(1-p)^{L-TF_T} \quad (1)$$

- It is maximum when $p = f$.

Experiment with a single document (or a subset of documents)

- Once we have for all terms their probabilities $p(t)$ of occurrence in an arbitrary piece of text we can make other experiments.

- Let d be a document and let $o_1, \ldots, o_l$ the experiment associated to the document. Among these we observe tf occurrences of t out of $l(d)$. Its frequency is $f = \frac{tf}{l(d)}$.

- What is the probability p of having t in the document, provided that the document is modelled by a Bernoulli's process?

$$P(p|f) = B(tf, l(d), p) = \binom{l(d)}{tf} p^{tf}(1-p)^{l(d)-tf} \tag{2}$$

Experiment with a document (or a subset of documents)

$$P(p|f) = B(tf, l(d), p) = \binom{l(d)}{tf} p^{tf} (1-p)^{l(d)-tf} \tag{3}$$

- A term is random in a document when the probability $P(p|f)$ is maximized, namely when $p = f$.

- Diverges from randomness when the probability $P(p|f)$ is *minimised*, namely when $f \gg p$.

- If a term is random in each document then it is a stop word (similarly to Harter's model)

A significant term in a document (or in an homogeneous sample of documents) does not follow the binomial law

- $B(tf, l(d), p)$ contains the information on how much significant is the term in the document.

- Instead of computing the probability of relevance $prob(q|d)$, as in the language model, we compute the *im*probability of obtaining this term as the document were assembled by a random process.

- we give a weight to the term which is inversely related to its probability of occurring under a random process.

Extracting significant term from a (small) set of documents:query expansion

- The *im*probability of having a term in a document by chance can be given by

$$Inf(t|d, D) = -\log B(tf, l(d), p) \tag{4}$$

- Example. Retrieve the set $T = \{d_1, d_2, d_3\}$ of the first 3 documents, from the query "What is a prime factor?"

term	tfq	tf	p	$Inf(t\|T, D)$	$nInf$	$tfq + 0.5 \cdot nInf$
prime	1	55	$6.4 \cdot 10^{-5}$	428.04	1.0000	1.5000
number	0	99	$1.4 \cdot 10^{-3}$	412.48	0.9636	0,4818
factor	1	49	$1.83 \cdot 10^{-4}$	299.67	0.7001	1,3500
integ	0	30	$4.36 \cdot 10^{-5}$	225.19	0.5261	0,2630
primal	0	8	$3.17 \cdot 10^{-6}$	76.77	0.1794	0,0896
multipl	0	15	$1.78 \cdot 10^{-4}$	68.74	0.1606	0,0802
test	0	21	$6.24 \cdot 10^{-4}$	68.28	0.1595	0,0797
divid	0	11	$6.28 \cdot 10^{-5}$	62.53	0.1461	0,0730
common	0	15	$2.65 \cdot 10^{-4}$	60.34	0.1410	0,0704
odd	0	9	$2.62 \cdot 10^{-5}$	60.26	0.1408	0,0703

The baseline for TREC-10 is about 21% of average precision. By adding this model of query expansion we get more than 25% of average precision (best run was 22.25%)

Why binomial law was not used before?

- Harter's work was about indexing. People were looking for term weighting functions, based on the assumption that indexing$\neq$ term-weighting.

- Harter assumed that documents had the same length (term frequency normalization was missing).

- Bernoulli's is a cumbersome formula to implement. I used different limiting formulas which are very good approximations.

- BM25 claims that is derived by the 2-Poisson model, and Poisson is an approximation of the Binomial law.

The use of the binomial law for weighting terms

- The improbability $Inf(t|d, D)$ is the initial step for computing the information content of a term in a document.

- The set of document E_t (the Elite set of a term) containing the term can be considered as an homogeneous piece of text. The distribution of "significant" terms in E_t deviates from that of a random process.

The aftereffect model in the Elite set E_t

- A rare event, like the occurrences of a word in a text, suddenly may become very frequent in a portion of text (e. g. a document).

- The observation of tf tokens of a term in a portion of a document increases our expectation of encountering the same word in the rest of the document.

- A large number of occurrences of a rare term in a small portion of text suggests us that the probability of encountering a new occurrence into an homogenous piece of text is almost certain.

$$p(tf + 1|tf, d, E_t) \sim 1$$

Urn models for modelling the aftereffect I

$$p(tf+1|tf,d,E_t) \sim 1$$

The risk is

$$1-p(tf+1|tf,d,E_t) \sim 0$$

the term weight is:

$$weight = gain = risk \cdot Inf(tf|d,D)$$

- Risk with Laplace's law of succession

$$risk = 1-p(tf+1|tf,d,E_t) \sim \frac{1}{tf+1}$$

Urn models for modelling the aftereffect II

$$p(tf+1|tf,d,E_t) \sim 1$$

The risk is

$$1-p(tf+1|tf,d,E_t) \sim 0$$

$$weight = gain = risk \cdot inf(tf|d,D)$$

- We add a new token of the word in the collection and we compute what is the probability of having this token in the document by chance (ratio of Bernoulli's processes). The risk is:

$$risk = \frac{B(tf+1,F_t+1,p)}{B(tf+1,F_t,p)} = \frac{F_t+1}{n_t(tf+1)}$$

  where n_t is the size of the Elite set and F_t the number of occurrences of t in the Elite set

The aftereffect model in summary

- We make a decision when observing a term within a document: either it is a good descriptor of the document or not.

- If we accept t as a document descriptor we take some risk. (as in Ponte–Croft's language model).

- We minimise the error by considering only the actual gain portion of the information content.

$$Inf(t|d, D) = gain + loss$$

$$weight = gain$$

Weighting terms

$$Inf(t|d, D) = gain + loss$$

$$weight = gain$$

- The risk of accepting a term is inversely related to its term frequency in the document with respect to the elite set

- The aftereffect model computes the conditional probability of having a new occurrence of the term once we have observed tf ones.

$$weight = gain = [1 - p(tf + 1|tf, d, E_t)] \cdot Inf(t|d, D)$$

Term frequency normalization

Binomial law treats documents as they were of the same length.

- The distribution is uniform

$$tfn = tf \cdot \frac{avg\_length}{l(d)}$$

- The term frequency distribution is a function of the length:

$$tfn = tf \cdot \log\left(1 + \frac{avg\_length}{l(d)}\right)$$

- tfn is given by the Zipfian like distribution

$$tfn = tf \cdot \left(\frac{avg\_length}{l(d)} \right)^A$$

Weighting formulas

$$weight = gain = risk \cdot Inf(t|d, D)$$

- We have 7 basic models to compute $Inf(t|d, D)$ (e.g. Bose-Einstein statistics)
- We have 2 models to compute the aftereffect (Polya's models of after-effect and theory of accidents should be still explored)
- We have 3 models for term frequency normalization.
- We have several models for query expansions.

- We have thus many basic models and also their combinations can improve results.

TREC 10, topics 501-550. Relevant documents: 3363

Unexpanded runs

Models	AvegPr	Pr5	Pr10	Pr20	R-Pr	Rel Ret
$I(n)B2$	0.2073	0.4120	0.3700	0.3120	0.2431	2377
$I(n)L2$	0.2031	0.4120	0.3540	0.3080	0.2354	2393
$I(n_e)B2$	0.2082	0.4120	0.3660	0.3170	0.2464	2406
$I(n_e)L2$	0.2016	0.4000	0.3600	0.3060	0.2380	2296
B_EB2	0.2084	0.4080	0.3720	0.3170	0.2464	2401
B_EL2	0.2014	0.4150	0.3580	0.3030	0.2358	2295
$PB2$	0.2036	0.3920	0.3440	0.2970	0.2349	2407
$PL2$	0.2093	0.4120	0.3640	0.3240	0.2423	2454

TREC 10, topics 501-550. Relevant documents: 3363
Expansion method: Bernoulli

Models	AvegPr	Pr5	Pr10	Pr20	R-Pr	Rel Ret
$I(n)B2$	0.2402	0.4200	0.3840	0.3110	0.2733	2522
$I(n)L2$	0.2573	0.4160	0.3920	0.3140	0.2797	2522
$I(n_e)B2$	0.2515	0.4400	0.3940	0.3150	0.2815	2528
$I(n_e)L2$	0.2406	0.4160	0.3900	0.3120	0.2670	2471
B_EB2	0.2497	0.4360	0.3960	0.3140	0.2804	2521
B_EL2	0.2379	0.4120	0.3820	0.3110	0.2664	2464
$PB2$	0.2152	0.3800	0.3420	0.2920	0.2464	2493
$PL2$	0.2372	0.4400	0.3800	0.3260	0.2757	2591
TREC-10 best run	0.2225	-	0.3440	0.2860	-	-

Modelling IR by computing the divergence from randomness

- It is a modular framework with 4 independent components. All models have an excellent performance.

- It is purely theoretical and we do not need to train the system.

- It is parameter free system. We do not need to learn or estimate parameters by using the bayesian methodology.

- The matching function is easy to implement. All models use at least 5 and at most 6 random variables which are provided by the statistics of the collection.

Research issues

- Term frequency normalization still need a systematic and foundational treatment. We have now working models but why them and not others?

- Many different and sophisticated models of aftereffect can be defined.

- Query expansion can be refined. Poor query expansion happens when the informative contents of the terms of the query are lower than those of added terms.

- Integration of the basic model with o ther random variables (proximity, co-occurrence)

- Integration with the link analysis.