Technology Review: BM25 and its Variant (BM25F and BM25+)

Ye Xu (yex2@illinois.edu)

1. Introduction

BM25 is one of the best-known and most successful instantiation of term weighting and document scoring function widely used in text retrieval industry (e.g. Search engine). It was introduced in the CS410 course as an example of vector space model because it consists of three key components that can be easily interpreted by heuristics: transformation of term frequency in a given document (TF), inverse document frequency (IDF) and the document length normalization. However, BM25 actually is a ranking algorithm based on the Probabilistic Retrieval Framework (PRF) [3] developed back in 1970s to estimate the document relevance to a given query. It was founded on the assumption of independent and binary relevance. In other words, document relevance is a property only dependent on the given information need without reference to other documents, and this property has only two status: relevant and not relevant [3][6]. Given this binary document by document property, BM25 was originally derived from the probability of the relevance, or the statement of the Probability Ranking Principle (PRP): Documents are ranked by their corresponding probability of relevance to given a query, *P(Rel |d, q)* [6]. This mini-review will first show how the BM25 ranking function is derived from the probabilistic model and then introduce two of its most popular variant BM25F and BM25+.

2. Derivation of BM25 as a probabilistic model

2.1. Basic model from the Probability Ranking Principle [3][6]

Given a query, we wish to rank the retrieved documents by the descending order of P(R = 1 | d, q) or the increasing order of P(R = 0|d,q) where R is relevance with = 1 means relevant = 0 means irrelevant, d means document and q means query. Therefore, we can rank the documents by taking the odds of these two probabilities and apply the Baye rules to get:

$$O(Rel\,|d,q) = \frac{P(R=1|d,q)}{P(R=0|d.q)} = \frac{P(d\,|R=1,q)P(R=1|q)}{P(d|R=0,q)P(R=0|q)}$$

As $\frac{P(R=1|q)}{P(R=0|q)}$ is constant for a given query, we only need to estimate the likelihood ratio of documents given relevance status and the query. We assume the words independence in the document for a given query and apply the Naïve Bayes Conditional Independence and logarithm to this ratio to get:

$$\frac{P(d|R=1,q)}{P(d|R=0,q)} \approx \prod_{V}\frac{P(tf_i|R=1,q)}{P(tf_i|R=0,q)} \approx \prod_{Q}\frac{P(tf_i|R=1,q)}{P(tf_i|R=0,q)} \propto \sum_{Q} log\frac{P(tf_i|R=1,q)}{P(tf_i|R=0,q)}$$

Where *tf_i* represents the frequency of term *i* in a document, V represents vocabulary and Q represents terms that appear in the query.

Let $U_i(tf_i) = log\frac{P(tf_i|R=1,q)}{P(tf_i|R=0,q)}$. As we are only interested in terms that appear in both the query and the document, the above equations can further evolve as:

$$\sum_{Q} U_i(tf_i) = \sum_{Q,tf_i>0} \left(U_i(tf_i) - U_i(tf_0)\right) + \sum_{Q} U_i(tf_0) \propto \sum_{Q,tf_i>0} \left(U_i(tf_i) - U_i(tf_0)\right)$$

Where *tf₀* represents that the term frequency is 0 in a document. Now we obtain the basic weighting function for each query term that appears in the document:

$$w_i = U_i(tf_i) - U_i(tf_0) = log\frac{P(tf_i|R = 1)P(tf_0|R = 0)}{P(tf_i|R = 0)P(tf_0|R = 1)} \quad (1)$$

2.2. Derivation of IDF component of BM25 weight from Binary Independence Model [5]

Similar to the Vector Space Model we learned from the course, we started to make binary assumption on the term frequency by letting tf_i = 1 (present in the document) or tf_i = 0 (absent in the document) for BM25, and turn equation (1) to:

$$w_i^{BIM} = log\frac{P(tf_i = 1|R = 1)(1-P(tf_i=1|R=0)}{P(tf_i=1|R=0)(1- P(tf_i=1|R=1)} \quad (2)$$

A reasonable way to estimate the four probability in the above equation is to use the maximal likelihood such that the probability of a binary status of a term frequency (tf_i = 1 or 0) given a relevance status of the document (*R* = 1 or 0) is approximated as the proportion of the judged document containing the judged term t_i. For example, $P(tf_i = 1|R = 1) = \frac{r_i}{R}$, where R is the total number of relevant documents and r_i is the number of such documents that contain the term i. To avoid negative or positive infinities by plugging this estimate directly into the above equation, a pseudo count of frequency, 0.5, was added to give the following equation:

$$w_i = log\frac{(r_i+0.5)(N-R-n_i+ r_i+0.5)}{(n_i- r_i+0.5)(R-r_i+0.5)} \quad (3)$$

Where R is the total number of relevant documents,

r_i is the number of relevant documents that contain the term *i*,
n_i is number of documents that contain the term *i*,
N is the total number of the documents.

However, even though we can make estimate of probabilities conditioned on relevant document (e.g. learn from user's feedback), it is usually impractical to estimate based on non-relevance. Thus, a further approximation was proposed by assuming any unknown document as non-relevant documents and setting R = r_i = 0. The above equation becomes:

$$w_i^{IDF} = log\frac{N-n_i+0.5}{n_i+0.5} \quad (4)$$

And this is the IDF component of BM25. Another way to interpret this is to apply information theory. Assuming we have ni documents that contain the query term i, then the probability of a randomly picked document that contain the term i will be $\frac{n_i}{N}$, and the corresponding information message that "Document D contains query term i " is $-log\frac{n_i}{N} = \log\frac{N}{n_i}$, which is very similar to the equation (4) above.

2.3. Derivation of BM25 weight precursor from the Eliteness model [2][3]

When learning the Vector Space Model in the lecture, we applied a non-linear transformation to the term frequency so that the contribution to the ranking by a term decreases as its occurs too often in the document. Same idea for BM25 was derived from the Eliteness model which introduced a latent property of "Eliteness" for the document- query term pair: If the term is "elite" in the document, then the content of the document is about the concept denoted by the term to some extent. Thus this "Eliteness" property associates the Relevance to the query and the term frequency shown in Figure 1 below:

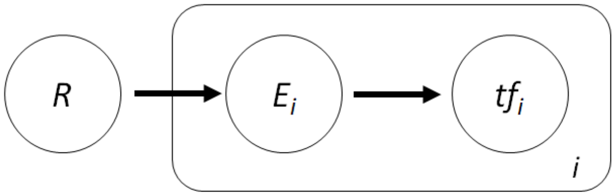

Figure 1: DAG of relation between Relevance to query (*R*), Eliteness (*E*) and term frequency (*tf*).

Assuming the Eliteness is also binary, then for query term *i* the relationship between Relevance and Eliteness can be indicated as: $p_{i1} = P(E_i = 1|R = 1)$ and $p_{i0} = P(E_i = 1|R = 1)$, and the relationship between term frequency and Eliteness can be $E_{i1}(tf) = P(tf_i|E_i = 1)$ and $E_{i0}(tf) = P(tf_i|E_i = 0)$. By plugging these to the basic model of the term weights of equation (1), we get the following equation:

$$w_i^{Elite} = log\frac{(p_{i1}E_{i1}(tf)+(1-p_{i1}E_{i0}(tf))(p_{i0}E_{i1}(tf_0)+(1-p_{i0}E_{i0}(tf_0))}{(p_{i1}E_{i1}(tf_0)+(1-p_{i1}E_{i0}(tf_0))(p_{i0}E_{i1}(tf)+(1-p_{i0}E_{i0}(tf))} \quad (5)$$

Which will be 0 if *tf = tf₀ = 0* and increase monotonically if *tf > 0* but asymptotically approaches a maximum limit which is the binary model of equation (2). Such an asymptotic behavior of the function $w_i^{Elite}(tf)$ adds a boundary of a term's contribution to the document ranking score just as we explained intuitively in the Vector Space Model. And the next step is to find a function with this saturation property, and the author of BM25 chose the following simple form:

$$\frac{tf}{k + tf} \quad (k > 0)$$

And by combining this saturation function with equation (4), we get the weight precursor of BM25 as follow:

$$w_i(tf) = \frac{tf}{k+tf}\, w_i^{IDF} \quad (6)$$

2.4. Full BM25 with Document Length Normalization [1][3]

Intuitively, the probability of the term frequency would be related to the length of the document. In most cases, the document lengths are not the same and the chance of seeing a query term would be higher with a long document than a shorter document. Thereby, the document scoring needs to be normalized by the document length so that it penalizes the long documents that are lengthy because they are verbose (e.g. using more words to describe the same idea) but doesn't do so to long documents with wider scope (e.g. having more content). So as we learned, a soft length normalizer is introduced as:

$$B = (1 - b) + b\frac{dl}{avdl}\ , \ \ 0 \le b \le 1 \quad (7)$$

Where *dl* is the length of the documents (total number of terms in the document), *avdl* is the average document length of the whole collection, and *b* is the modifier of the normalization level: normalization is full when *b=1* and is off when *b=0*. In BM25, this normalizer is applied to the *tf* before the *tf* transformation by the saturation function such that $tf' = \frac{tf}{B}$ and by replacing *tf* by *tf'* in equation 6, the final form of BM25 term weight is:

$$w_i^{BM25} = \frac{tf'}{k+tf'} w_i^{IDF} = \frac{tf}{k\left(1-b+b\frac{dl}{avdl}\right)+tf} w_i^{IDF} \quad (8)$$

There are several versions of the BM25 that are slightly variant from the original form. One common variant is to multiply a *(k + 1)* to the numerator of the saturation function so that it will be more compatible with the IDF component. Furthermore, the IDF component also has variant implementations. The one in equation (4) is derived from the binomial model and has issues when more than half of the documents in the collection contains the term. In this situation the IDF component $w_i^{IDF} = log\frac{N-n_i+0.5}{n_i+0.5}$ is negative. If we have two almost identical document, then the one that contains the term will have lower rank than the one does not. So this component varied in different ways to solve this problem. The formula we learned in the course is:

$$w_i^{BM25} = \frac{tf(k+1)}{k\left(1-b+b\frac{dl}{avdl)}\right)+tf} \times log\frac{N+1}{n_i} \quad (9)$$

And the final form of BM25 by applying this version of weight to the term frequency give us:

$$\sum\nolimits_{Q,tf_i>0} tf \times \frac{tf(k+1)}{k\left(1-b+b\frac{dl}{avdl)}\right)+tf} \times log\frac{N+1}{n_i} \quad (10)$$

3. BM25F for structured document [2][3]

3.1. Search in structured document

Practically, documents are not a single undifferentiated body of text but are usually structured into multiple fields. For example, a research paper consists of title, abstract, main body, and reference. Some of these fields tend to be more informative or predictive than others. A query that match the title of a research paper may provide stronger support that a match that fit into the body. Thereby, given a ranking algorithm such as BM25, one can improve the relevance retrieval by applying the algorithm separately to each field of the document and then combine them with weighted linear combination, and this is the basic idea of BM25F.

3.2. BM25F

Given a document that consists of a set of fields, *1* to *F*, each field *j* will have an assigned weight v_j, field length is fl_j and the frequency of term *i* in filed j is tf_{ji}. Therefore, a modified version of BM25 by replacing the *tf* in equation (10) with its equivalence, the field, gives rise to:

$$w_i^{BM25F} = \frac{\widehat{tf_{J\iota}}}{k\left(1-b+b\frac{\widehat{dl}}{\widehat{avdl}}\right)+\widehat{tf_{J\iota}}} w_i^{IDF} \quad (11)$$

Where $\widehat{tf_\iota} = \sum\nolimits_{field} v_j tf_{ji}$, $\widehat{dl} = \sum\nolimits_{field} v_j\, fl_j$ and $\widehat{avdl} = \frac{1}{N}\sum\nolimits_{j=1:N} \widehat{dl_J}$.

The above simple form can be further improved by allowing parameters of BM25 such as the normalizer *B* vary between fields such that for each field *j*:

$$B_j = \left(1 - b_j\right) + b_j \frac{sl_j}{avsl_j}\ ,\ 0 \le b_j \le 1\ (12)$$

By normalizing $\widehat{tf_{ji}}$ by B_j the same way to derive equation (8), a formula of BM25F weighting function for the term *i* is:

$$w_i^{BM25F} = \frac{\widehat{tf_{ji}}'}{k+\widehat{tf_{ji}}'}\, w_i^{IDF}\ \ (13)$$

Where $\widehat{tf}'_{ji} = \sum_{field} v_j \frac{tf_{ji}}{B_j}$.

4. BM25+ for improvement of document length normalization on term frequency [4]

A common deficiency of the standard BM25 algorithm is that the normalization of term frequency *tf* by document length is not lower bounded properly so that very long document tends to be overly penalized. As shown in equation (7), when the document length *dl* is very large, the pivoted document length normalizer B becomes very large and the normalized $tf' = \frac{tf}{B}$ is approaching 0. As a result, the ranking score of such a long document is approaching 0 as if the query term does not occur in that document at all. This will cause the problem that the occurrence of a term in the long document cannot ensure a higher ranking of such a document than documents that do not contain the term but are shorter. In fact, such a deficiency is also observed for other retrieval algorithms including PL2 method, Dirichlet Prior method, and pivoted normalization method.

To fix this problem, two constraints were added to the standard BM25 model to capture two heuristics that (1) there is sufficient gap between presence and absence of a term in the document which should not be closed by the document length normalization. If two documents received the same relevance by matching all other query terms except for one term that occurs only in document X but not in document Y, then document X should be scored higher than Y. (2) Documents that cover more distinctive terms should be rewarded sufficiently. If the query includes term *a* and *b*, document X covers *a* twice and *b* once, then it should be ranked higher than document Y that covers term *a* three times but does not contain *b*. To implement these, the BM25+ adds an additional parameter δ and the formula of weighting function is:

$$w_i^{BM25+} = [\frac{tf(k+1)}{k\left(1-b+b\frac{dl}{avdl)}\right)+tf} + \ \delta]\ w_i^{IDF}\ \ (14)$$

Where δ takes default value of 1 without any training data.

5. Implementation of BM25

Some high-level coding languages have packages implementing BM25 and its variants. Rank_bm25 package from the PyPi project (https://pypi.org/project/rank-bm25/) provides implementation of Okapi BM25, BM25L, BM25+, BM25-Adpt, BM25T. MatLab also provides function bm25similarity (https://www.mathworks.com/help/textanalytics/ref/bm25similarity.html) and by choosing arguments between 'DocumentLengthScaling' and 'DocumentLengthCorrection', one can apply BM25+ and other calculations with different constraints on the document length normalization.

6. Summary

The original Okapi BM25 was first implement for information retrieval system back in 1980's based on the classic probabilistic models. Through years of development based on empirical and theoretical analysis, it has evolved slightly variant from its original formula such that some part was simplified (e.g. the non-linear transformation of the query term frequency *qtf*), some problem was fixed (e.g. lower-bounding problem of *tf* normalization by document length by BM25+) and the application was extended (e.g. BM25F). But it still holds the fundamental retrieval signals that are shared by all effective retrieval models: the term frequency (*tf*), the inverse document frequency (IDF) and document length. State of art algorithms like BM25 all rely on the good combination and interaction of these three components that address the problem and meet the practical needs in text retrieval industry most comprehensively and effectively.

Reference:

(1) Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In Proceedings of ACM SIGIR 1996.

(2) S. E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval, Proceedings of ACM SIGIR 1994.

(3) S. Robertson and H. Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond, Found. Trends Inf. Retr. 3, 4 (April 2009).

(4) Y. Lv, C. Zhai, Lower-bounding term frequency normalization. In Proceedings of ACM CIKM 2011.

(5) https://en.wikipedia.org/wiki/Binary_Independence_Model

(6) C. D. Manning, P. Raghavan and H. Schütze. Introduction to Information Retrieval. https://nlp.stanford.edu/IR-book/html/htmledition/irbook.html.