IR Assignment 3

Shivam Chaturvedi

March 2015

1 Introduction

1.1 Question 1 a

Rocchio's method starts with the equation for $\vec{q}_{opt}$ as

$$\vec{q}_{opt} = \underset{\vec{q}}{\text{argmax}}[sim(\vec{q}, C_r) - sim(\vec{q}, C_n)] \tag{1}$$

where C_r = Set of relevant documents
C_n = Set of non-relevant documents

Using the cosine similarity formula as:

$$cos(\theta) = \frac{A \cdot B}{|A||B|} \tag{2}$$

(1) can be rewritten as:

$$\vec{q}_{opt} = \underset{\vec{q}}{\text{argmax}} \left[\frac{\vec{q} \cdot C_r}{|\vec{q}||C_r|} - \frac{\vec{q} \cdot C_n}{|\vec{q}||C_n|} \right] \tag{3}$$

$$= \underset{\vec{q}}{\text{argmax}} \left[\frac{\vec{q} \cdot \sum\limits_{d_j \in C_r} d_j}{|\vec{q}||C_r|} - \frac{\vec{q} \cdot \sum\limits_{d_j \in C_n} d_j}{|\vec{q}||C_n|} \right] \tag{4}$$

$$= \underset{\vec{q}}{\text{argmax}} \left[\frac{\vec{q} \cdot \sum\limits_{d_j \in C_r} d_j}{|\vec{q}||C_r|} - \frac{\vec{q} \cdot \sum\limits_{d_j \in C_n} d_j}{|\vec{q}||C_n|} \right] \tag{5}$$

Intuitively, the equation in the brackets will be maximum when the query, $\vec{q}$ has values such that it contains the summation of the terms in the relevant terms and removes all the terms in summation of the terms in the non-relevant terms.

This means that the unit vector $\hat{q} = \frac{\vec{q}}{|q|}$ should be aligned such that

$$\hat{q} \cdot \sum_{d_j \in C_r} d_j = \sum_{d_j \in C_r} d_j \tag{6}$$

and

$$\hat{q} \cdot \sum_{d_j \in C_n} d_j = \sum_{d_j \in C_n} d_j \tag{7}$$

Hence, we can write the equation for $\vec{q}_{opt}$ as:

$$q_{opt} = \frac{1}{|C_r|} \sum_{d_j \in C_r} d_j - \frac{1}{|C_n|} \sum_{d_j \in C_n} d_j \tag{8}$$

1.2 Question 1 b

Yes, I agree with the statement that a straight-forward application of the Rocchios algorithm for query reformulation will lead to an inefficient algorithm in practice. This is basically because of 2 important reasons:

1. Too many added terms tend to retrieve too many non-relevant documents which may increase overall recall, but definitely worsen precision.
2. Subsequent reformulation can remove important terms from the reformulated query too.

The most viable solution to deal with the above issues is to weigh the existing and the new added terms and the terms to be removed so that there is a gradual shift in the query and the reformulation shifts the influence of terms and does not perform extreme operations once and for all.

1.3 Question 2 a

The Binary Independence Retrieval Model (BIRM) tries to calculate the probability of the document being relevant with respect to a specific query q_k Now, we can try to calculate the odds of this event (that the document with set of terms d_m is relevant to q_k. We can take the document vector d_m to be a binary vector $\vec{x}$ such that:

$$x_i = \begin{cases} 1, \text{if } t_i \in d_m \\ 0, \text{otherwise} \end{cases} \tag{9}$$

Now, Odds of the document vector can be used for ranking by calculating:

$$O(R|q_k, \vec{x}) = \frac{P(R=1|q_k, \vec{x})}{P(R=0|q_k, \vec{x})} \tag{10}$$

$$= \frac{P(R=1|q_k, \vec{x})}{P(R=0|q_k, \vec{x})} \tag{11}$$

$$= \frac{P(R=1|q_k)}{P(R=0|q_k)} \cdot \frac{P(\vec{x}|R=1, q_k)}{P(\vec{x}|R=0, q_k)} \tag{12}$$

BIR model makes an independence assumption under which the following holds true:

$$\frac{P(\vec{x}|R=1,q_k)}{P(\vec{x}|R=0,q_k)} = \prod_{i=1}^{n} \frac{P(x_i|R=1,q_k)}{P(x_i|R=0,q_k)} \tag{13}$$

What the above means in simpler terms is that BIRM assumes that there is no dependence of one terms on the other terms in the document. Hence their probabilities can be taken independently and taken product of.

Now, from equation (12), we see that the first term does not depend on the document. So we can omit this term as we are considering ranking the documents with respect to the odds.

Then, using the independence assumption from (13), (12) can be re-written as:

$$O(R|q_k,\vec{x}) = \prod_{i=1}^{n} \frac{P(x_i|R=1,q_k)}{P(x_i|R=0,q_k)} \tag{14}$$

Now, we can separate the terms where $x_i = 0$ and $x_i = 1$ because they are already independent (as assumed) and there are only 2 values that x_i can take.

$$O(R|q_k,\vec{x}) = \prod_{x_i=1}^{n} \frac{P(x_i|R=1,q_k)}{P(x_i|R=0,q_k)} \cdot \prod_{x_i=0}^{n} \frac{P(x_i|R=1,q_k)}{P(x_i|R=0,q_k)} \tag{15}$$

Now, let $p_i = P(x_i = 1|R,q)$ and $q_i = P(x_i = 0|R,q)$. So, (15) can be written as:

$$O(R|q_k,\vec{x}) = \prod_{x_i=1}^{n} \frac{p_i}{q_i} \cdot \prod_{x_i=0}^{n} \frac{1-p_i}{1-q_i} \tag{16}$$

Adding two new terms to the equation as following:

$$O(R|q_k,\vec{x}) = \prod_{x_i=1}^{n} \frac{p_i}{q_i} \cdot \prod_{x_i=0}^{n} \frac{1-p_i}{1-q_i} \cdot \prod_{x_i=1}^{n} \frac{1-q_i}{1-p_i} \cdot \prod_{x_i=1}^{n} \frac{1-p_i}{1-q_i} \tag{17}$$

$$= \prod_{x_i=1}^{n} \frac{p_i}{q_i} \cdot \prod_{x_i=1}^{n} \frac{1-q_i}{1-p_i} \cdot \prod_{x_i=0}^{n} \frac{1-p_i}{1-q_i} \cdot \prod_{x_i=1}^{n} \frac{1-p_i}{1-q_i} \tag{18}$$

$$= \prod_{x_i=1}^{n} \frac{p_i(1-q_i)}{q_i(1-p_i)} \cdot \prod_{q_i=1}^{n} \frac{1-p_i}{1-q_i} \tag{19}$$

where we can see that in equation (18), the last two terms product up and make all the terms for which $q_i = 1$ i.e. which are present in the query term and the document. Hence, they can be clubbed together.

We can again see that this combined last terms does not really contribute to the ranking of the document as it is not dependent on the document, but on the query. Hence, it can also be ignored for ranking purposes.

$$O(R|q_k, \vec{x}) = \prod_{x_i=1} \frac{p_i(1-q_i)}{q_i(1-p_i)} \tag{20}$$

Taking log of the values:

$$O(R|q_k, \vec{x}) = \log \prod_{x_i=1} \frac{p_i(1-q_i)}{q_i(1-p_i)} \tag{21}$$

$$= \sum_{x_i=1} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \tag{22}$$

where the term on the right is also called the Retrieval Status Value (RSV_d)

$$\text{RSV}_d = \sum_{x_i=1} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \tag{23}$$

$$= \sum_{x_i=1} c_i \tag{24}$$

$$\text{where } c_i = \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \tag{25}$$

$$= \log \frac{p_i}{1-p_i} + \log \frac{1-q_i}{q_i} \tag{26}$$

Now, we can reasonably make the following estimates:

$$p_i \approx 0.5 \text{Harper and Croft assumption of distributing terms evenly.} \tag{27}$$

Hence, the first term reduces to 0 and 2nd term:

$$\log \frac{1-q_i}{q_i} = \log \frac{N-df_i}{df_i} = \log \frac{N}{df_i} = \text{IDF} \tag{28}$$

Hence, we see that the weights actually amount to the IDF of each document.

Assumption. The primary assumption that BIRM make is the independence of the terms. This is not so true in realistic environments where one word might depend on the presence of the other term, such as "President Obama". If we only have "President" then it would refer to any President, but with "Obama", it makes sense to match the US president. Another example can be "exam policy". Without the "exam", policy can refer to any policy such as a bank policy. In such situations, the model will fail.

1.4 Question 2 b

The Probability Ranking Principle or PRP denotes a binary notion of relevance. W know that a document can be relevant or non-relevant to the query. PRP

states that we rank the documents by their estimated probabilities of relevance with respect to the information need i.e. by $P(R = 1|d, q)$ where d is the document vector and q is the query vector of terms.

Using PRP and the given values, the expected outcome will be 1–2–3. This is the best ranking that is possible, because as we can see from the covariance, 1 and 2 are slightly positively correlatd, and we know that 1 is definitely relevant (0.5), we can say that 2 is also relevant, but with lower ranking. 3 is negatively correlated with 1, and hence might not be relevant.

—END OF DOCUMENT—