CS 510 Assignment 2

Daniel Campos

March 14th,2021

1 Probabilistic Retrieval Model

1.1 Multinominal Ranking

1.1.1 Show that ranking if a document is relevant to a query is equivalent to the sum of the probability of each word being relevant to the query

First off, recapping the RSJ model where $score(Q, D) \overset{\text{rank}}{=} \sum_{i=1,d_i=q_i=1}^{k} \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$. We know that a multi-variate Bernoulli models focuses on the presence and absence of a feature where for IR this feature is occurrence in the document. We also know that a Multinomial model leverages the number of counts of a feature, commonly referred to as term frequency. Essentially, a Bernoulli model is the same as a multinomial model where all frequencies have been simplified to 1. Thus the first step is to introduce the term frequency into the calculation which gives $score(Q, D) \overset{\text{rank}}{=} \sum_{i=1,d_i=q_i=1}^{k} c(w, D) \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$. We then amplify the summation from all terms in the query present in the document ($\sum_{i=1,d_i=q_i=1}^{k}$) to all the words in the vocabulary ($\sum_{w \in V}$) since we can no longer just focus on term occurrence but now have to focus on frequency. Since we have now are modeling the frequency of a feature instead of its absence/existence we drop the normalization of $(1 - q_i)$ and $(1 - p_i)$. Thus we arrive at our target formula $score(Q, D) \overset{\text{rank}}{=} \sum_{w \in V} c(w, D) \log \frac{p(w|Q,R=1)}{p(w|Q,R=0)}$

1.1.2 How many parameters are in such a model

2: $p(w|Q, R = 1)$ for $w \in V$, and $p(w|Q, R = 0)$ for $w \in V$

1.2 Give the formula for Maximum likelihood estimate of $p(w|Q, R = 0)$

$p(w|Q, R = 0) = \sum_{D \in C} \frac{tf_w}{|D|}$ where tf_w is the amount of times a word w occurs in document D and $|D|$ is the document length.

1.3 Give the formula for Maximum likelihood estimate of $p(w|Q, R = 1)$ where we use the query as the only relevant document

$p(w|Q, R = 1) = \frac{tf_w}{|Q|}$ where tf_w indicates the occurrence of word w in the query Q and $|Q|$ represents the query length.

1.4 Give the formula for smoothing the maximum likelihood estimate using Jelinek-Mercer with the collection language model

$p(w|Q, R = 1) = (1 - \beta)\frac{tf_w}{|Q|} + \beta p(w|C)$ where tf_w indicates the occurrence of word w in the query Q and $|Q|$ represents the query length, β is a Jelinek-Mercer smoothing coefficient and $p(w|C)$ is the language model unigram probability.

1.5 Write Down the retrieval function. Does the retrieval function capture TF, IDF, document length normalization?

$score(Q, D) \overset{\text{rank}}{=} \sum_{w \in V} c(w, D) \log \frac{(1-\beta)\frac{tfq_w}{|Q|} + \beta p(w|C)}{\sum_{D \in C} \frac{tfd_w}{|D|}}$ where tfq_w is the term frequency in the query Q and tfd_w is the terfm frequency in document D, $|Q|$ represents the query length,$|D|$ is the document length, β is a Jelinek-Mercer smoothing coefficient and $p(w|C)$ is the language model unigram probability.
This captures TF because we take the term probability and multiply it by the frequency, it captures IDF because we normalize the probability of the word in Query given how common the word is in the corpus and it captures document length normalization because each of our estimations normalize frequency by query/document length.

2 Language Models

2.1 Show that if we use query likelihood scoring method with Jelinker-Mercer smoothing function we can rank documents with $score(Q, D) \overset{\text{rank}}{=} \sum_{w \in Q \cap D} c(w, Q) \log(1 + \frac{(1-\lambda)c(w,D)}{\lambda p(w|C) * D})$

This ranking function is ranking documents by assigning documents score by the recall of query terms. First, the sum $\sum_{w \in D \cap D}$ means this will rank documents that have matches on query terms higher than any without. This is necessary to remove any noise that non matched by high probability terms may have for document score. Next, this function multiplies the probability by the count of word occurrence in the query, $c(q, Q)$ which serves to add higher importance to common query terms. Finally the ranking function includes a measure of term relevance in the given document. The numerator $(1 - \lambda)c(w, D)$ models the term occurrence in the document. The denominator $\lambda p(w|C) * |D|$ represents the expected occurrence of the term in the document given its commonality in the corpus since $p(w|C)$ is overall unigram occurrence which is multiplied by document length $|D$ to provide an expected amount of term occurrences in document. This means that documents that have higher than the corpus wide expected term occurrence(and thus more relevant) have a higher probability than those that have a less than corpus average occurrence.

2.2 Vector Space

2.2.1 What would be the query vector

The query vector in this case is represented by $\sum_{w \in Q \cap D} c(w, Q)$ as we can essentially consider it a one hot vector for query document term matches. Since we only sum over terms that match all unmatched terms are set to 0. Since we multiply by $c(w, Q)$ any term that is not 0(and thus occurs) is set to its occurrence in the query.

2.2.2 What would be the document vector

The document vector is the $\log(1 + \frac{(1-\lambda)c(w,D)}{\lambda p(w|C)*D})$ portion of the summation. Terms that occur in both the document and query are given the value of their occurrence in the document vs their expected occurrence in the document.

2.2.3 What is the similarity function

The similarity function is the multiplication by query vector and the document vector. Terms that occur in the document more than the LM predicts make a document more relevant than terms that occur less. These probabilities are summed over the multiplied query and document vector to produce a similarity score.

2.2.4 Does the term weight in the document vector capture TF-IDF weighting and the document length normalization heuristics?

It captures all 3 heuristics because of the document vector model. TF is captured by $c(w, D)$, IDF is captured by the use of the LM probability in the denominator$(p(w|C))$ and document length normalization is captured by the multiplication of the IDF by document length$(p(w|C) * |D|)$.

2.3 Are artificially long documents penalized using Jelinek-Mercer and Dirichlet prior smoothing?

In Jelinek-Mercer artificially long documents are not penalized because the smoothing function is static everywhere in the prediction. In other words if we increase the document by size k the scores will remain the same. On the other hand Dirichlet prior smoothing actually prefers longer documents since there is a separation in the normalization of document length $(n \log \frac{\mu}{|d|+\mu}$ and the term occurrence $(\log(1 + \frac{c(w,d)}{\mu p(w|C)}$ and thus is we increase the document length by a factor of k our first term grows by a factor of k which places more importance on it than the document length

3 KL-Divergence

3.1 Show that KL Divergence covers the query likelihood retrieval function if the language model is set to the word distribution in the query

Kullback-Leibler divergence (or relative entropy) is written as $D(P|Q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$. If we build on the assumption that $p(w|\theta_q) = \frac{c(w,Q)}{|Q|}$ to represent word likelihood given a query and $p(d|\theta_D) = \frac{c(w,D)}{|D|}$ then we can treat θ_q and θ_D as estimates of the query and document language models then the relevance can be measured by the negative KL divergence. We use the negative KL divergence because items that have a lower entropy between q and d are more relevant. Our query likelihood function becomes $score(Q, D) \overset{\text{rank}}{=} -D(\theta_Q|\theta_D) = \sum_w p(w|\theta_Q) \log p(w|\theta_D) + (-\sum_w p(w|\theta_q) \log p(w|\theta_q))$. The second term is a query dependent constant which is ignored for ranking.

4 Divergence Minimization Feedback

We start off with out optimization problem $f(\theta*) = \arg\min[\frac{1}{n}(\sum_{i=1}^{n} D(\theta||\theta_{e_i}) - \lambda D(\theta||\theta_C$.

Using the fact that $g(p_1, p_2, ..., p_n) = \sum_{w \in V} p(w|\theta) = 1$ we use the Lagrange multiplier to find the point of minimum entropy across all probability distributions, p' which we do by requiring that $\frac{\delta}{\delta p'}(f + \lambda(g-1))| = 0$.

This gives a system of λ equations such that $\frac{\delta}{\delta p_k}(-(\sum_{j=1}^{l} ambdap_j \log(p_j)) + \lambda(\sum_{j=1}^{\lambda} -1)) = 0$. Carrying out the differentiation we get $-(\frac{1}{\ln 2} + log_2 p'_k) + \lambda = 0$ which shows that all p'_k are equal because they depend only on λ.

By then using our constraint that $\sum_{w \in V} p(w|\theta) = 1$ we find that $p'_k = \frac{1}{\lambda}$ meaning that the uniform distribution is that uniform with the most entropy and thus maxima. As a result the minima happens at $p'_k = \frac{1}{1-\lambda}$

Substituting this information in, $p(w|\theta) \alpha exp(\frac{1}{1-\lambda}\frac{1}{|n|}\sum_{i=1}^{n} D(\theta||\theta_{E_i})) - \frac{\lambda}{1-\lambda} D(\theta||\theta_C)$.

Knowing that $D(\theta||\theta_C) = \log p(w|C)$ and $D(\theta||\theta_{E_i}) = \log p(w|\theta)$ we rewrite to our final form. $p(w|\theta) \alpha exp(\frac{1}{1-\lambda}\frac{1}{|n|}\sum_{i=1}^{n} \log(p|\theta_i) - \frac{\lambda}{1-\lambda} \log p(w|C))$