CS 510 Advanced Information Retrieval

MP 1

Problem 1:

(a)

Given $p(w \mid \theta_D) = \begin{cases} p_s(w \mid \theta_D) \;\; if\ w \in D \\ \alpha_D p(w \mid C)\ otherwise \end{cases}$ ---- (1)

Now, for JM smoothing, we know that $P(w \mid D)$ = (1-λ) $\frac{c(w,D)}{|D|}$ + $\lambda\, P(w|C)$

where $P(w|C)$ is the probability of the word 'w' in the corpus 'C' , $c(w, D)$ is the count of the word 'w' in document 'D' and λ is the smoothing parameter that controls the weight to be assigned to the unseen words in the document.

- Now, consider the case if $w \in D$, then $p_s(w \mid D) = P(w \mid D)$ = (1-λ) $\frac{c(w,D)}{|D|}$ + $\lambda\, P(w|C)$

(comparing with equation (1))

Hence, $p_s(w \mid D)$ = (1-λ) $\frac{c(w,D)}{|D|}$ + $\lambda\, P(w|C)$ ---- (A)

- Now, consider the case if w $\notin D$, then c(w, D) = 0, and hence $P(w \mid D) = \lambda\, P(w|C)$

Comparing with the given equation (1), we see that for w $\notin D, P(w \mid D) = \alpha_D P(w|C)$

Hence, we have $\alpha_D = \lambda$ ---- (B)

(b)

Given $p(w \mid \theta_D) = \begin{cases} p_s(w \mid \theta_D) \;\; if\ w \in D \\ \alpha_D p(w \mid C)\ otherwise \end{cases}$ ---- (1)

Now, for DP smoothing, we know that $P(w \mid D) = \frac{|D|}{\mu+|D|} \frac{c(w,D)}{|D|} + \frac{\mu}{\mu+|D|}\, P(w|C)$

where $P(w|C)$ is the probability of the word 'w' in the corpus 'C', $c(w, D)$ is the count of the word 'w' in document 'D' and μ is the smoothing parameter that controls the weight to be assigned to the unseen words in the document.

- Now, consider the case if $w \in D$, then $p_s(w \mid D) = P(w \mid D) = \frac{|D|}{\mu+|D|} \frac{c(w,D)}{|D|} + \frac{\mu}{\mu+|D|} P(w|C)$

(comparing with equation (1))

Hence, $p_s(w \mid D) = \frac{|D|}{\mu+|D|} \frac{c(w,D)}{|D|} + \frac{\mu}{\mu+|D|} P(w|C)$ ---- (A)

- Now, consider the case if w $\notin D$, then c(w, D) = 0, and hence $P(w \mid D) = \frac{\mu}{\mu+|D|} P(w|C)$

Comparing with the given equation (1), we see that for w $\notin D, P(w \mid D) = \alpha_D P(w|C)$

Hence, we have $\alpha_D = \frac{\mu}{\mu+|D|}$ ---- (B)

(c)

Given $p(w \mid \theta_D) = \begin{cases} p_s(w \mid \theta_D) & if\ w \in D \\ \alpha_D p(w \mid C) & otherwise \end{cases}$ ---- (1)

Now, for two stages smoothing, we are given that,

$P(w \mid D)$ = (1- λ) $\frac{c(w,D)+ \mu P(w|C)}{\mu+|D|}$ + $\lambda\, P(w|C)$

Rearranging, we get:

$P(w \mid D)$ = (1- λ) $\frac{|D|}{\mu+|D|} \frac{c(w,D)}{|D|}$ + (1- λ) $\frac{\mu}{\mu+|D|} P(w|C) + \lambda\, P(w|C)$

where $P(w|C)$ is the probability of the word 'w' in the corpus 'C', $c(w, D)$ is the count of the word 'w' in document 'D' and μ is the smoothing parameter that controls the weight to be assigned to the unseen words in the document.

- Now, consider the case if $w \in D$, then

$p_s(w \mid D) = P(w \mid D)$ = (1- λ) $\frac{|D|}{\mu+|D|} \frac{c(w,D)}{|D|}$ + (1- λ) $\frac{\mu}{\mu+|D|} P(w|C) + \lambda\, P(w|C)$

(comparing with equation (1))

Hence, $p_s(w \mid D)$ = (1- λ) $\frac{|D|}{\mu+|D|}$ $\frac{c(w,D)}{|D|}$ + (1- λ) $\frac{\mu}{\mu+|D|}$ $P(w|C) + \lambda\, P(w|C)$ ---- (A)

- Now, consider the case if w $\notin D$, then c(w, D) = 0, and hence

$P(w \mid D)$ = (1- λ) $\frac{\mu}{\mu+|D|}$ $P(w|C) + \lambda\, P(w|C)$ = ((1- λ) $\frac{\mu}{\mu+|D|}$ + λ) $P(w|C)$

Comparing with the given equation (1), we see that for w $\notin D, P(w \mid D) = \alpha_D P(w|C)$

Hence, we have $\alpha_D =$ (1- λ) $\frac{\mu}{\mu+|D|}$ + λ ---- (B)

Problem 6:

(a)

Looking at the plots, we see that two types of keyword queries, namely short and long keyword queries behave similarly and the two types of verbose, namely short and long verbose queries behave similarly.
The plots suggest that the sensitivity to smoothing parameters is generally low in the case of the keyword queries, long or short than for the verbose queries, long or short.

The table below contains the **range of MAP attained** by each of the methods for each of the four query types for both datasets.

From the plots, it is clear that even the short verbose queries are more sensitive to the changes in smoothing parameters than the long keyword queries.
Hence, we can see that the sensitivity to smoothing parameters is more correlated with the verbosity of the query than with the length of the query. This is because insufficient smoothing is much harmful for verbose queries than for keyword queries because in verbose queries, insufficient smoothing is leading to explaining the common word of the query more.

Hence, MAP fluctuates more when we smoothen with varying smoothing parameters, the short or long verbose queries than when we smoothen the long and short keyword queries. Giving less weight to the unseen words than is required in case of verbose queries is similar to giving more weight to the frequent words, that occurred due to verbosity of the query and hence the MAP varies compared to cases when we smoothen more.

Since a concise keyword query has less or no non-informative common words, the effect of smoothing can be expected to be much more dominated by explaining the unobserved words in a document. But for a verbose query containing high ratio of non-informative common words, the effect of explaining the common and non-informative words in a query will have more influence.
Hence, verbose queries are more sensitive to the changes in smoothing parameter than the keyword queries.

	Dataset ap88-89			**Dataset ziff1-2**		
Query Type	**JM**	**DP**	**Two stage**	**JM**	**DP**	**Two stage**
Short Keyword	0.01	0.02	0.03	0.01	0.03	0.01
Long Keyword	0.01	0.04	0.05	0.02	0.04	0.02
Short Verbose	**0.06**	**0.06**	0.02	**0.04**	**0.04**	0.02
Long Verbose	**0.04**	**0.06**	0.05	**0.05**	**0.06**	0.02

(b)

Looking at the plots, we see that two types of keyword queries, namely short and long keyword queries behave similarly and the two types of verbose queries, namely short and

long verbose queries behave similarly for both JM and Drichlet Prior smoothing method. The shape of the short and long keyword queries looks similar. The shape of the short and long verbose queries looks similar. The shape of both of the keyword queries differs from the shape of the verbose queries.

Also, we can notice that the sensitivity to smoothing parameters is generally low in the case of the keyword queries, long or short than for the verbose queries, long or short. i.e. the keyword query plots fluctuate less as we vary the smoothing parameter but verbose query plots fluctuate more as we change the value of the smoothing parameter.

Also, we can notice a consistent order of performance for all the queries in each plot. We can see that the long keyword queries give the highest MAP values for both JM and Drichlet Prior smoothing methods, followed by long verbose query MAP values, then short keyword queries and the short verbose queries give the least mean average precision across both the methods and datasets.
This means that queries with only keywords tend to perform better than more verbose queries and that the longer queries generally give higher mean average precisions than short queries.

(c)

In two-stage smoothing, where Jelinek-Mercer is used to do the smoothing for the second time after Dirichlet prior smoothing, the sensitivity pattern through plots is much more similar for keyword queries and verbose queries. **We see a more consistent and stable behavior now.**
Hence, we may say in case of two stage smoothing, the sensitivity pattern due to smoothing parameter is less dependent to the query type and length.
Also notice that the optimal value for the prior parameter is roughly at the same point for each of the four different query types in this case. So, we may conclude that there are no differences in the shape of the plots between query types for two stage smoothing method.

There are two different roles for smoothing i.e explaining the unobserved words in a document, and explaining the common and non-informative words in a query.
For any query, the effect of two stage smoothing is a mixed effect of both roles of smoothing for JM as well as Dirichlet prior smoothing.
For a keyword query, the effect can be expected to be much more dominated by explaining the unobserved words in a document, since keyword query has less or no noninformative words.
But for a verbose query, the effect of explaining the common and non-informative words in a query will have more influence since verbose queries generally have a higher ratio of noninformative words.
Since Dirichlet prior method adapts to the document length naturally, which is desirable for explaining the unobserved words in a document, in general DP smoothing performs the best on keyword queries.
Also, since in Jelinek-Mercer smoothing, we set a fixed smoothing parameter across all documents, which is desirable for explaining the non-informative words in a query, in general JM smoothing performs the best on verbose queries.

Now since in two-stage smoothing, we first use a Dirichlet prior to smoothen the language model and then in the second stage use the JM smoothing, the effect of both explaining the unobserved words in a document and explaining the non-informative words in a query are incorporated. **Hence,**

the sensitivity pattern (shape) through plots is much more similar for keyword queries and verbose queries in two stage smoothing method.

(d)

Since Dirichlet prior method adapts to the document length naturally, which is desirable for explaining the unobserved words in a document, in general DP smoothing performs the best on keyword queries.
Also, since in Jelinek-Mercer smoothing, we set a fixed smoothing parameter across all documents, which is desirable for explaining the non-informative words in a query, in general JM smoothing performs the best on verbose queries.

The table below contains the maximum value of MAP attained by each of the methods for each of the four query types for both datasets.

	Dataset ap88-89			Dataset ziff1-2		
Query Type	**JM**	**DP**	**Two stage**	**JM**	**DP**	**Two stage**
Short Keyword	0.220427	0.242113	0.23887910	0.13468654	0.15918	0.1588875
Long Keyword	0.3496304	0.3562506	0.35446596	0.2039725	0.212567	0.2109322
Short Verbose	0.21485085	0.2298183	0.23098625	0.12018423	0.137753	0.1475278
Long Verbose	0.3057243	0.3123652	0.31211234	0.18687750	0.2008172	0.2040375

The above claim can be seen through the values in the table above as well that JM works best for verbose queries and DP works best for the keyword queries.

Now, in two stages smoothing with the help from Jelinek-Mercer, Dirichlet Prior smoothing can achieve a higher MAP on verbose queries than just with DP smoothing alone. Also, the two stage results are better than the best result of using Jelinek-Mercer alone. We see a more consistent and stable behavior now. Also, in case of two stage smoothing, the sensitivity pattern due to smoothing parameter is less dependent to the query type and length.

Thus, the two-stage smoothing performance appears to be the best performing single method at its optimal parameter setting for any query type. Hence, I will choose Two stage smoothing method for the search engine, if the query type is unknown.
Next, with two-stage smoothing, the optimal setting of JM method is correlated with the query while that of Dirichlet prior method is correlated to the document collection.
Hence, it is possible to optimize μ on only the document collection basis without depending on queries and similarly optimize λ only based on queries.
So, for the optimal parameters, I can look for optimal λ in JM method applied alone to both datasets and optimal μ in DP method applied alone to both datasets and use them as the optimized pair for the two stages smoothing model.

Notice that $\lambda = 0.7$ give the optimal value for JM smoothing (or 0.8 mostly) applied alone for queries and $\mu = 2000$ give the optimal value for DP smoothing (or 1000 mostly) applied alone for queries.

Hence, for the search engine, I will choose two stage smoothing method with $\boldsymbol{\lambda = 0.7}$ and $\boldsymbol{\mu = 2000}$ to get the optimal results, when type of the query is unknown.

Submitted by: Rachneet Kaur, NET ID: rk4