A Markovian Approach to Evaluate Session-based IR Systems
Electronic Appendix

David van Dijk[1], Marco Ferrante[2][0000−0002−0894−4175], Nicola Ferro[2][0000−0001−9219−6239], and Evangelos Kanoulas[1][0000−0002−8312−0694]

University of Amsterdam, The Netherlands {d.v.van.dijk@hva.nl, e.kanoulas@uva.nl}
University of Padua, Italy {ferrante, ferro}@dei.unipd.it

A Pseudocode for the Computation of the MsM Measure

Algorithm 1 stems from the observation that, since the random variables in eq. (1) can take the value ∞ with positive probability, their expectation is equal to ∞. Nevertheless, in order to measure the "stochastic distance" of the document at position (i, j) from that in $(1, 1)$, we can compute the expectation under the restriction that the search does not end before reaching the target document. In practice, we can equal to 0 the probability to reach the state F from the states above (i, j) and normalize the other elements to still have a stochastic matrix. Therefore, in the algorithm, we can use a single smaller transition matrix P, representing the dynamic within a generic query, with $N + 2$ states: the first N states represents the N retrieved documents within the query, the $F = N + 1$ state is the end of the search session, and the $Q = N + 2$ is the reformulation to the next query. We then, appropriately, select rows and columns from this transition matrix P, rescale them into a $\hat{P}$ matrix ensuring it is stochastic, and perform the different computations, as discussed in the pseudo-code.

Algorithm 1: Pseudo-code for computing the MsM measure (It continues on the next page.)

Input: run a $N \times K$ integer matrix where rows are documents, columns are queries, and each cell contains a natural number representing the relevance of a document, 0 for not relevant. i is the document index, j is the query index.

Input: p, q, r, and s, the transition probabilities of, respectively: moving forward to the next document within a query; moving backward to the previous document within a query; moving to the next query, i.e. jumping to the first document of the next query; and, stopping a search session. We assume $p + q + r + s = 1$, $p > 0$, $q \geq 0$, $r > 0$, and $s > 0$.

Output: the MsM measure for the input run.

Data: For the first document of a generic query, there is no backward transition probability, so we have to rescale the transition probabilities: $p_1 = \frac{p}{p+s+r}$, $s_1 = \frac{s}{p+s+r}$, $r_1 = \frac{r}{p+s+r}$

Data: For the last document of a generic query, there is no forward transition probability, so we have to rescale the transition probabilities: $q_N = \frac{q}{q+s+r}$, $s_N = \frac{s}{q+s+r}$, $r_N = \frac{r}{q+s+r}$

Data: The transition matrix P of a generic query is a $(N+2) \times (N+2)$ stochastic matrix where:

- the leading $N \times N$ principal sub-matrix contains the backward and forward transitions between the documents within a query, paying attention to the re-scaled transition probabilities for the first and last document;
- the additional absorbing state $F = N + 1$ represents the end of a search session;
- the additional absorbing state $Q = N + 2$ represents the jump to the next query.

$$
P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \\ N \\ F \\ Q \end{array}
\begin{array}{c}
\begin{array}{cccccccc|cc} 1 & 2 & 3 & 0 & \cdots & N-2 & N-1 & N & F & Q \end{array} \\
\left(\begin{array}{cccccccc|cc}
0 & p_1 & 0 & 0 & \cdots & 0 & 0 & 0 & s_1 & r_1 \\
q & 0 & p & 0 & \cdots & 0 & 0 & 0 & s & r \\
0 & q & 0 & p & \cdots & 0 & 0 & 0 & s & r \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & q & 0 & p & s & r \\
0 & 0 & 0 & 0 & \cdots & 0 & q_N & 0 & s_N & r_N \\
\hline
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1
\end{array}\right)
\end{array}
$$

Data: e is a $N \times 1$ vector, representing the average time to go from document 1 to documents $1, \ldots, N$, respectively.

Data: e_Q is a scalar, representing the average time to move to the next query.

Data: h_F is a scalar, representing the probability to go from document 1 to state F, i.e. end of search session, in the first $K - 1$ queries.

Data: h_{FK} is a scalar, representing the probability to go from document 1 to state F, i.e. end of search session, in the K-th query.

Data: h is a $1 \times K$ vector, representing the probability to go from document 1 of the first query to the state $(F, 1), \ldots, (F, K)$, respectively.

Data: π is a $1 \times K$ vector, representing the probability of ending the search session in query $1, \ldots, K$, respectively.

Data: g is a $1 \times K$ vector, representing the total gain of each query.

```
/* the average time to go from document 1 to itself is 0          */
```
$e(1) \leftarrow 0$;

Algorithm 1: Pseudo-code for computing the MsM measure. (Continued from the previous page and it continues on the next page.)

```
/* P̂_1 is the same matrix as P but where the F and Q state rows
   and columns have been removed since, to compute the average time
   to move from document 1 to document i, you must assume that
   neither the search session has ended nor you have moved to the
   next query                                                    */
```
$\hat{P}_1 \leftarrow P(1:N, 1:N)$;

ensure $\hat{P}_1$ is a stochastic matrix, i.e. normalize each row so that it sums up to 1;

for $i \leftarrow 1$ **to** $N-1$ **do**

```
    /* compute the average time to go from each document to document
       i+1 (I_i is the i×i identity matrix)                          */
```
$$tmp \leftarrow \left(\mathbf{I}_i - \hat{P}_1(1:i, 1:i)\right)^{-1} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix};$$

```
    /* we are interested only in the average time to go from
       document 1 to document i+1                                    */
```
$e(i+1) \leftarrow tmp(1)$;

end

```
/* P̂_2 is the same matrix as P but where the F state rows and
   columns have been removed since, to compute the average time to
   move to the next query, you must assume that the search session
   has not ended                                                 */
```
$\hat{P}_2 \leftarrow P([1:N\ Q], [1:N\ Q])$;

ensure $\hat{P}_2$ is a stochastic matrix, i.e. normalize each row so that it sums up to 1;

```
/* compute the average time to go from each document to the next
   query                                                         */
```
$$tmp \leftarrow \left(\mathbf{I}_N - \hat{P}_2(1:N, 1:N)\right)^{-1} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix};$$

```
/* we are interested only in the average time to go from document 1
   to the next query                                             */
```
$e_Q \leftarrow tmp(1)$;

```
/* compute the probability to go from document i to state F in the
   first K-1 queries                                             */
```
$tmp \leftarrow \left(\mathbf{I}_N - P(1:N, 1:N)\right)^{-1} P(1:N, F)$;

```
/* we are interested only in the probability to go from document 1
   to state F in the first K-1 queries                           */
```
$h_F \leftarrow tmp(1)$;

```
/* P̂_3 is the same matrix as P but where the Q state rows and
   columns have been removed since, to compute the probability to
   go from document i to state F in the K-th query, there is not
   next query                                                    */
```
$\hat{P}_3 \leftarrow P(1:F, 1:F)$;

ensure $\hat{P}_3$ is a stochastic matrix, i.e. normalize each row so that it sums up to 1;

Algorithm 1: Pseudo-code for computing the MsM measure. (Continued from the previous page, last page.)

/* compute the probability to go from document i to state F in the K-th query */

$tmp \leftarrow \left(\mathbf{I}_N - \hat{P}_3(1:N, 1:N)\right)^{-1} \hat{P}_3(1:N, F)$;

/* we are interested only in the probability to go from document 1 to state F in the K-th query */

$h_{FK} \leftarrow tmp(1)$;

/* compute the probability to go from document 1 (of the first query) to state F of the j-th query */

for $j \leftarrow 1$ **to** $K-1$ **do**

  $h(j) \leftarrow (1 - h_F)^{j-1} \cdot h_F$;

end

$h(K) \leftarrow (1 - h_F)^{K-1} \cdot h_{FK}$;

/* compute the probability of ending the search session in query j */

$\pi(1) \leftarrow 1$;

for $j \leftarrow 2$ **to** K **do**

  $\pi(j) \leftarrow 1 - \sum_{l=1}^{j-1} h(l)$;

end

/* discount each retrieved document by the average time to get to it */

for $i \leftarrow 1$ **to** N **do**

  for $j \leftarrow 1$ **to** K **do**

    $E(i,j) \leftarrow \frac{run(i,j)}{1+e(i)+e_Q}$;

    /* In case of logarithmic discount, use instead $E(i,j) \leftarrow \frac{run(i,j)}{1+\log_{10}(1+e(i)+e_Q)}$; */

  end

end

/* compute the total gain for each query */

for $j \leftarrow 1$ **to** K **do**

  $g(j) \leftarrow \pi(j) \sum_{i=1}^{N} E(i)$;

end

/* compute the MsM measure */

$MsM \leftarrow \sum_{j=1}^{K} g(j)$;