Score Normalization Using Logistic Regression with Expected Parameters

Robin Aly

Human Media Interaction, University Twente, 7522AE Enschede, The Netherlands
r.aly@utwente.nl

Abstract. State-of-the-art score normalization methods use generative models that rely on sometimes unrealistic assumptions. We propose a novel parameter estimation method for score normalization based on logistic regression, using the expected parameters from past queries. Experiments on the Gov2 and CluewebA collection indicate that our method is consistently more precise in predicting the number of relevant documents in the top-n ranks compared to a state-of-the-art generative approach and another parameter estimate for logistic regression.

1 Introduction

Search engines rank documents by scores that were designed to express "higher is better" and vary wildly between queries. Some applications however benefit from scores reflecting the probability of relevance, e.g. to model the number of documents a user should read from a ranked list [4] or to fuse results of different search engines [5]. The main challenge for score normalization methods that transform scores into such probabilities is to find precise and robust transformation functions for the scores of a given query. In this paper we propose a novel score normalization method based on logistic regression that uses results from previous queries.

State-of-the-art score normalization methods model the probability of relevance by making assumptions about the shapes of the density functions of scores in relevant and non-relevant documents [3]. These methods face the challenge that the actual scores sometimes violate the assumed shapes [2]. We normalize scores using logistic regression, which model the relationship between scores and probabilities as a sigmoid function, i.e. without prior modeling of density functions. Logistic regression models make weaker assumptions about scores and use less parameters, which might be the reason why Arampatzis and Robertson [3] recently referred to them as "an under-explored avenue worth pursuing".

Nottelmann and Fuhr [13], who were among the first to use logistic regression in score normalization, learn parameters from relevance judgments of a set of training queries. This approach has the disadvantage that it learns a transformation between scores to the probability of relevance to *any* of these queries. In [1], we found that such probabilistic models do not conform to the probability of relevance principle because they are not specific to the current query. We therefore

M. de Rijke et al. (Eds.): ECIR 2014, LNCS 8416, pp. 579–584, 2014.

approach parameter estimation by learning parameters for each training query individually by using the expected parameter values from training queries. Our experiments show that this method improves accuracy even with limited training material. Additionally, the expected parameter values can be seen as a naive query specific estimate and therefore have potential to be improved in future work.

The rest of this paper is structured as follows: in Section 2 we describe related work to this paper. Section 3 describes our ranking framework for news items. The experiments which conducted to evaluate our approach is described in Section 4. We finish this paper with conclusions and proposals for future work in Section 5.

2 Related Work

Arampatzis and Robertson [3] recently provide an exhaustive survey over the vast body of literature on score normalization methods that produce probabilities of relevance: most normalization methods use generative approaches based on score distributions in relevant documents and non-relevant documents. For example, Gamma distributions [11] or an exponential distribution for scores in non-relevant documents and a normal distribution for scores in relevant documents [12] have been tried. [4] consider a truncated normal distribution for scores in relevant documents, which is more realistic than normal distributions, which have infinity support. Arampatzis and Robertson [3] conclude that a pair of exponential-normal distributions is currently the best performing. We use logistic regression models for score normalization, which only assume that the probability of relevance increases monotonically with scores. Additionally, logistic regression models are mathematically less complex compared to generative methods and require less parameters (two parameters versus four in generative approaches).

Nottelmann and Fuhr [13] are among the first to investigate logistic regression for score normalization, which has been considered before as a ranking function, see for example Cooper et al. [6]. Nottelmann and Fuhr assume a single logistic regression model that transforms scores of arbitrary queries to probabilities of relevance. They estimate the corresponding parameter setting from the scores of judged documents for a set of training queries. As a result, the transformation function for typical scores over multiple queries. However, scores have been found to differ between queries and our experiments indicate that this approach sometimes produces poor performance. Our estimation approach generates for each query a parameter setting, which we combine to the expected parameters assuming queries are random samples from a population. Our estimate is therefore similar to a prior estimate for the parameters for a query at hand that does not consider any query-specific data.

Note that there are also score normalization methods that produce other quantities than probabilities of relevance. For example, [10] consider the commutative score distributions of historical queries and Arampatzis and Kamps [2] consider the part of a signal in a noisy channel as normalized scores. Although these methods do sometimes achieve strong performance, they are only distantly related to our method and will not be treated further here.

3 Estimating Logistic Models

We now describe our method to transform scores into probabilities of relevance using logistic regression. Because logistic regression models are seldom used in information retrieval, we first provide a brief introduction and we refer the interested reader to [13] for a more in-depth discussion. In the score normalization scenario, logistic regression assume that the probability of relevance R for a document with score s is defined as:

$$P(R|s) = \frac{1}{1 + exp(-w_1 - w_2\ s)} \tag{1}$$

where $\boldsymbol{w} = (w_1, w_2)$ are the model parameters. This notation does not specify the exact probabilistic model used. As explained in [1], the model could belong to Model 2, which consider the relevance of documents to a specific queries, or to Model 0, which considers the relevance between multiple queries and documents. Nottelmann and Fuhr [13] model probabilities of relevance for Model 1 because they use as training data a set of relevance judgments between a set of queries $\mathcal{Q}$ and the documents in the collection $X = \{(r_{q,d}, s_{q,d})\} \forall q \in \mathcal{Q}$, where $r_{q,d} \in \{0, 1\}$ is the relevance status and $s_{q,d}$ is the scores between query q and document d. They use the maximum likelihood estimate $\boldsymbol{w}_n$ given the data X:

$$\boldsymbol{w}_n = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P_{\mathcal{Q}}(X|\boldsymbol{w})$$

where the $P_{\mathcal{Q}}$ is the probability measure for the probability of relevance to *any* query in $\mathcal{Q}$ with a score s. For a new query q these parameters are used in equation Eq. (1) to calculate the probability of relevance.

In contrast to this estimation procedure, our method considers for each training query $q \in \mathcal{Q}$ the data $X_q = \{(r_{q',d}, s_{q',d}) \in X | q' = q\}$, where X is defined above. For each training query, we calculate the maximum likelihood parameter estimate $\boldsymbol{w}_q$ given the data X_q:

$$\boldsymbol{w}_q = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P_q(X_q|\boldsymbol{w}) \tag{2}$$

where P_q is the measure for the probability of relevance for the query q. We consider $\boldsymbol{w}_q$ as optimal values of a random variable $\boldsymbol{W}$ defined on the sample space of queries $\mathcal{Q}^+$. Now, when a retrieval system encounters a new query $\hat{q}$, we cannot calculate the parameters $\boldsymbol{w}_{\hat{q}}$ by Eq. (2) as there are no relevance judgments available. Instead, we assume that $\hat{q}$ is random sample from $\mathcal{Q}^+$, and use the expected parameter values in query space $\mathcal{Q}^+$, $E_{\mathcal{Q}^+}[\boldsymbol{W}]$, as an unbiased estimate of for $\boldsymbol{w}_{\hat{q}}$. This expectation can be estimated by the mean of the parameters in the training queries:

$$\boldsymbol{w}_{\hat{q}} := E_{\mathcal{Q}^+}[\boldsymbol{W}] \simeq E_{\mathcal{Q}}[\boldsymbol{W}] = \left(\frac{\sum_q w_{q,1}}{|\mathcal{Q}|}, \frac{\sum_q w_{q,2}}{|\mathcal{Q}|} \right) \tag{3}$$

where the last last term are the mean in the training query set and $(w_{q,1}, \boldsymbol{w}_{q,2})$ are the parameters of training query q, which we determined through Eq. (2). Note that although this estimate is clearly coarse and equal for all new queries, it estimates parameters for individual queries and therefore estimates Model 2 probabilities.

Table 1. Results: mean absolute error in the given evaluation query set $\mathcal{Q}^t$. Column heads indicated the cut-off value n. $*$ indicates statistical significance compared to TruncExpNorm according to a paired t-test with $p = 0.05$.

Estimator Type	10	30	50	100	1000
TruncExpNorm	4.29	13.29	21.64	40.85	323.63
LogNottelman	4.70	14.79	26.42	58.37	812.87
LogExpectation	3.11*	8.66*	14.52*	26.72*	118.18*

(a) Gov2 $\mathcal{Q}^t = 701 - 750$

Estimator Type	10	30	50	100	1000
TruncExpNorm	3.77	11.65	19.18	38.26	329.92
LogNottelman	3.85	13.29	24.42	56.09	809.29
LogExpectation	2.62*	9.30	16.11	32.84	163.03*

(b) Gov2 $\mathcal{Q}^t = 751 - 800$

Estimator Type	10	30	50	100	1000
TruncExpNorm	4.94	13.65	20.25	34.57	317.10
LogNottelman	4.80	14.80	26.20	59.90	869.60
LogExpectation	3.04*	7.95*	12.60*	25.15*	130.65*

(c) Gov2 $\mathcal{Q}^t = 801 - 850$

Estimator Type	10	30	50	100	1000
TruncExpNorm	3.58	9.20	13.96	30.77	327.34
LogNottelman	6.76	19.96	35.72	75.21	701.30
LogExpectation	2.39*	7.44	10.19*	17.48*	70.85*

(d) ClueWebA $\mathcal{Q}^t = 001 - 50$

Estimator Type	10	30	50	100	1000
TruncExpNorm	3.09	9.74	15.25	28.17	347.08
LogNottelman	7.03	20.65	35.32	73.92	662.67
LogExpectation	2.54	7.97	12.90	20.28	60.56*

(e) ClueWebA $\mathcal{Q}^t = 051 - 100$

Estimator Type	10	30	50	100	1000
TruncExpNorm	3.31	8.57	16.83	32.12	324.84
LogNottelman	4.88	16.12	30.12	68.40	893.46
LogExpectation	3.84	9.51	16.94	32.18	156.66*

(f) ClueWebA $\mathcal{Q}^t = 101 - 150$

4 Experiments

We evaluated our approach on two web datasets: Gov2 (25M documents) using the TREC terrabyte track queries 1-150 and ClueWebA (250M documents after removal of the documents with an above median spam score [7]) using the TREC web track queries 700-850. To balance between the number of training queries and evaluation queries, we consider queries in batches of 50 as test queries ($\mathcal{Q}^t$) and the remaining 100 queries as training queries ($\mathcal{Q}$) according to the year that they were used in TREC. In order to compare our method against the state-of-the-art, we implemented the generative model using truncated exponential and normal distributions by Arampatzis et al. [4] (TruncExpNorm) and the logistic regression model with the estimation method by Nottelmann and Fuhr [13] (LogNottelmann). We refer to the method presented here as LogExpectation. As the score function s we used Indri with default settings [1], which produces negative scores. As all methods require positive scores, we followed [4] and cut the original ranking at rank 1000 and added the smallest score to the scores of each document. Because LogNottelmann showed poor performance with these scores, we also divided the scores for this method by the maximum score, ensuring that

[1] `http://www.lemurproject.org/indri/`

they were between 0 and 1. For logistic regression training we used the libLinear software package [9]. Similar to the evaluation of the TREC Legal Track [8], we used the mean absolute error ME_n of the expected number of relevant documents and the actual number, which can be defined as:

$$ME_n = \frac{1}{|Q^t|} \sum_{q \in Q^t} \left| R_q^n - \sum_{i=1}^{n} P(R|s(d_i)) \right|$$

where R_q^n is the number of relevant documents in the top n of query q and $P_q(R|s(d_i))$ is the probability of relevance of the ith document d_i using the parameter estimates from Eq. (3). Note that a lower mean error is better.

Table 1 shows the results of our experiments. Our method has a lower mean error when estimating the number of relevant documents at almost all given cut-off values. The improvements are stronger for higher cut-off values. Due to space limitations, we refer the reader to the table for more detailed results. Note that we considered

5 Conclusions

We proposed a logistic regression model for score normalization, which requires fewer parameters and makes milder assumptions than state-of-the-art generative models. Compared to the method by Nottelmann and Fuhr [13], which uses logistic regression models to estimate parameters of a probability measure for all queries, our method uses the expected weights from sample queries as a constant estimate for a probability measure for individual queries. Because we recently found in [1] that probability measures for all queries do not conform to the probability ranking principle, our estimation method is therefore a first step towards solutions that obey this principle.

Using the Gov2 and Cluweb9A datasets with three query batches of 50 queries per dataset, we evaluated our method against a state-of-the-art generative model and the logistic regression model by Nottelmann and Fuhr [13]. Similar to the TREC Legal Track we used the mean error when estimating the number of relevant documents in the top-n ranks as an evaluation measure. Our method consistently improved the evaluation measure in both datasets for all considered query batches and was often statistically significant.

This work is among the first to use logistic regression models for score normalization. While the proposed method already achieves improvements compared to strong baselines, we believe that future work can leverage further improvements by adapting parameter estimates for each query.

Acknowledgments. The work reported in this paper was partially funded by the EU Project AXES (FP7-269980).

References

[1] Aly, R., Demeester, T., Robertson, S.: Probabilistic models in ir and their relationships. Information Retrieval, 1386–4564 (2013) ISSN 1386-4564, http://dx.doi.org/10.1007/s10791-013-9226-3, doi:10.1007/s10791-013-9226-3

[2] Arampatzis, A., Kamps, J.: A signal-to-noise approach to score normalization. In: CIKM 2009, USA. ACM (2009) ISBN 978-1-60558-512-3

[3] Arampatzis, A., Robertson, S.E.: Modeling score distributions in information retrieval. Information Retrieval 14, 1–21 (2010) ISSN 1386-4564

[4] Arampatzis, A., Kamps, J., Robertson, S.: Where to stop reading a ranked list?: threshold optimization using truncated score distributions. In: SIGIR 2009, USA, pp. 524–531. ACM (2009) ISBN 978-1-60558-483-6

[5] Callan, J.: Distributed information retrieval. In: Croft, W. (ed.) Advances in Information Retrieval. The Information Retrieval Series, vol. 7, pp. 127–150. Springer US (2000) ISBN 978-0-7923-7812-9, http://dx.doi.org/10.1007/0-306-47019-5_5, doi:10.1007/0-306-47019-5_5

[6] Cooper, W., Chen, A., Gey, F.C.: Experiments in the probabilistic retrieval based on staged logistic regression. In: TREC 1994. NIST (1994)

[7] Cormack, G.V., Smucker, M.D., Clarke, C.L.A.: Efficient and effective spam filtering and re-ranking for large web datasets. CoRR, abs/1004.5168 (2010)

[8] Cormack, G.V., Grossman, M.R., Hedin, B., Oard, D.W.: Overview of the trec 2011 legal track. In: TREC 2011, p. 1 (2011)

[9] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: Liblinear: A library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)

[10] Fernández, M., Vallet, D., Castells, P.: Using historical data to enhance rank aggregation. In: SIGIR 2006, USA. ACM (2006) ISBN 1-59593-369-7

[11] Kanoulas, E., Dai, K., Pavlu, V., Aslam, J.A.: Score distribution models: assumptions, intuition, and robustness to score manipulation. In: SIGIR 2010, USA, pp. 242–249. ACM (2010) ISBN 978-1-4503-0153-4

[12] Manmatha, R., Rath, T., Feng, F.: Modeling score distributions for combining the outputs of search engines. In: SIGIR 2001, USA, pp. 267–275. ACM (2001) ISBN 1-58113-331-6

[13] Nottelmann, H., Fuhr, N.: From uncertain inference to probability of relevance for advanced ir applications. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 235–250. Springer, Heidelberg (2003)