Notes on the SPUD Language Model

Ronan Cummins

Abstract

Recently, the Smoothed Pólya Urn Document (SPUD) language model was shown to outperform the multinomial language model for ad hoc information retrieval over a number of different types of collections for queries of various lengths. In this short note we outline very simply how to estimate the parameters of the model.

1 Introduction

The recently developed SPUD language model [1] treats document generation using the Pólya process. It has been shown to incorporate a number of theoretically interesting properties. For example, it models the scope and verbosity hypothesis [4] separately, and reintroduces a measure closely related to inverse document frequency [5]. This short note does not delve further into these issues, rather it aims to clearly outline how the parameters of the model can be estimated in a practical manner. This is aimed at the information retrieval practitioner who wishes to adopt a state-of-the-art unigram model for retrieval purposes.

2 Ranking

In short, each document d is assumed to have been drawn from a document model $\mathcal{M}_d$. Documents are ranked based on the likelihood of their model generating the query string q. It is assumed that query-terms are drawn with replacement from the model so that documents are ranked as follows:

$$log\ p(q|\mathcal{M}_d) = \sum_{t \in q} (log\ p(t|\mathcal{M}_d) \cdot c(t,q)) \tag{1}$$

where q is the query string, $\mathcal{M}_d$ is the document model, and $c(t,q)$ is the number of times term t appears in the query. This is the familiar query-likelihood approach [3].

3 Estimation

In this section we outline how to estimate the parameters of the model. First, Table 1 outlines notation needed for the understanding of the subsequent formulae.

Table 1: Notation

Key	Description
$c(t, d)$	frequency of term t in document d
$c(t, q)$	frequency of term t in query q
$\|d\|$	length of document d (i.e. number of tokens)
$\|\vec{d}\|$	length of document vector (# of distinct terms in document d)
df_t	document frequency (number of documents in which t occurs)
$\|q\|$	length of query q (i.e. number of tokens)
n	number of documents in the collection
$\|v\|$	vocabulary of the collection (# of distinct terms in the collection)

3.1 Document Model

We assume that each document is generated from a multivariate Pólya distribution $\boldsymbol{\alpha}_d$, also known as the Dirichlet-compound multinomial. The maximum-likelihood estimates of a document model are as follows:

$$\hat{\boldsymbol{\alpha}}_d = (m_d \cdot \frac{c(t_1, d)}{|d|}, m_d \cdot \frac{c(t_2, d)}{|d|}, ..., m_d \cdot \frac{c(t_{|v|}, d)}{|d|}) \tag{2}$$

where we set $m_d = |\vec{d}|$ to fully specify the model, which is the number of *word types* in the document. One of the main advantages of using the SPUD model is that it distinguished between *word types* and *word tokens* when normalising a document with respect to length.

3.2 Background Model

In order to overcome the *zero-probability* problem (i.e. over-fitting), the document model is smoothed with a background model. It has been shown that close approximations [2] to the maximum likelihood estimates of the multivariate Pólya distribution are as follows:

$$\hat{\boldsymbol{\alpha}}_c = (m_c \cdot \frac{df_{t_1}}{\sum_{t'} df_{t'}}, m_c \cdot \frac{df_{t_1}}{\sum_{t'} df_{t'}}, ..., m_c \cdot \frac{df_{t_1}}{\sum_{t'} df_{t'}}) \tag{3}$$

where m_c can be solved using numerical methods. The following iterative procedure can be used to estimate m_c:

$$m_c^{new} = \frac{\sum_j^n |\vec{d_j}|}{\sum_j^n \psi(|d_j| + m_c) - n \cdot \psi(m_c)} \tag{4}$$

where $\psi(x) = \frac{d}{dx} log\Gamma(x)$ is the digamma function and Γ is the gamma function and n is the number of documents in the collection. Experiments suggest that initialising m_c to the average document length is sufficient to enable the procedure to converge within 15 iterations even for large collections. This estimation can be done once off-line and is not computationally expensive in practice.

3.3 Linear Combination

Given the estimates of a document model $\boldsymbol{\alpha}_d$ and the background model $\boldsymbol{\alpha}_c$, we linearly smooth these using one free hyper-parameter ω as follows:

$$\mathcal{M}_d = (1 - \omega) \cdot \boldsymbol{\alpha}_d + \omega \cdot \boldsymbol{\alpha}_c \tag{5}$$

where empirical evidence suggests that $\omega = 0.8$ is a stable setting. The resultant $|v|$-dimensional vector is our document representation (or document model). This vector is a multivariate Pólya and so the expected value (a multinomial) of this distribution can be used as the point-estimate. It is relatively easy to see that the entire mass of each document model is $(1-\omega) \cdot m_d + \omega \cdot m_c$. Therefore, the final ranking formula is as follows:

$$\text{SPUD}_{dir}(q, d) = \sum_{t \in q} (log(\frac{(1 - \omega) \cdot |\vec{d}| \cdot \frac{c(t,d)}{|d|} + \omega \cdot m_c \cdot \frac{df_t}{\sum_j^n |\vec{d_j}|}}{(1 - \omega) \cdot |\vec{d}| + \omega \cdot m_c}) \cdot c(t, q)) \tag{6}$$

which can be further re-written in a computationally more efficient manner which is in a somewhat similar form to the multinomial language model [6]:

$$\text{SPUD}_{dir}(q, d) = |q| \cdot log(\frac{\mu'}{\mu' + |\vec{d}|}) + \sum_{t \in q \cap d} (log(1 + \frac{|\vec{d}| \cdot c(t, d) \cdot \sum_j^n |\vec{d_j}|}{\mu' \cdot |d| \cdot df_t}) \cdot c(t, q)) \tag{7}$$

where μ' can be substituted as follows:

$$\mu' = \frac{\omega}{1 - \omega} \cdot m_c \tag{8}$$

4 State-of-the-Art Retrieval

The SPUD model has recently been shown to outperform the state-of-the-art multinomial language model. Here we will step through the stages involved in converting documents to their probabilistic representations.

4.1 Estimation Example

Consider the document collection in Figure 1 with the documents represented as vectors of their frequency counts.

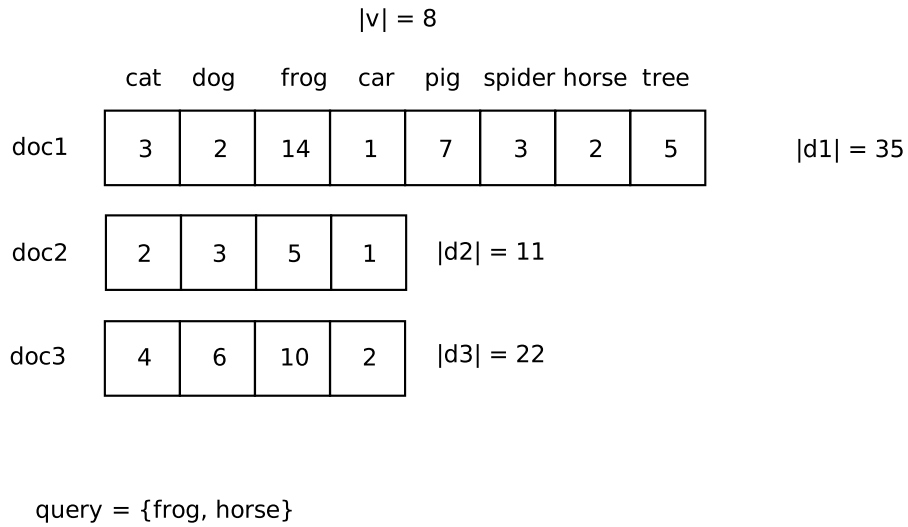

Figure 1: Sample collection of three documents

Given these three documents, we estimate the unsmoothed multivariate Pólya document model for each document (from Equation 2) and then smooth it with the background model (from Equation 3). The unsmoothed document models are estimated respectively as follows:

$$\boldsymbol{\alpha}_{d_1} = 8 \cdot < \frac{3}{35}, \frac{2}{35}, \frac{14}{35}, \frac{1}{35}, \frac{7}{35}, \frac{3}{35}, \frac{2}{35}, \frac{5}{35} > \tag{9}$$

$$\boldsymbol{\alpha}_{d_2} = 4 \cdot < \frac{2}{11}, \frac{3}{11}, \frac{5}{11}, \frac{1}{11} > \tag{10}$$

$$\boldsymbol{\alpha}_{d_3} = 4 \cdot < \frac{4}{22}, \frac{6}{22}, \frac{10}{22}, \frac{2}{22} > \tag{11}$$

where the attentive reader will see that the vectors for document 2 and document 3 are identical. The background model is estimated as follows:

$$\boldsymbol{\alpha}_c = 2 \cdot < \frac{3}{16}, \frac{3}{16}, \frac{3}{16}, \frac{3}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16} > \tag{12}$$

where the denominator of the expected multinomial is the sum of the number of non-zero dimensions of each document (i.e. $8 + 4 + 4 = 16$). This leaves m_c to be estimated using the iterative procedure in Equation 4 which converges to

$m_c \approx 2$ for this toy example. The background model is linearly combined with each document model using the hyper-parameter $\omega = 0.8$ such that all vectors have the same number of non-zero dimensions. This results in the following document model for each document:

$$\mathcal{M}_{d_1} = 3.2\cdot < 0.13, 0.12, \underline{0.29}, 0.10, 0.13, 0.07, \underline{0.05}, 0.11 > \quad (13)$$

$$\mathcal{M}_{d_2} = 2.4\cdot < 0.19, 0.22, \underline{0.27}, 0.16, 0.04, 0.04, \underline{0.04}, 0.04 > \quad (14)$$

$$\mathcal{M}_{d_3} = 2.4\cdot < 0.19, 0.22, \underline{0.27}, 0.16, 0.04, 0.04, \underline{0.04}, 0.04 > \quad (15)$$

The expected multinomial of these smoothed vectors can be used to rank the documents with respect to a query. For example, given the query {*frog, horse*}, the probability that this query was generated from $E[\mathcal{M}_{d_1}]$ is $0.29 \times 0.05 = 0.0145$ and for both $E[\mathcal{M}_{d_2}]$ and $E[\mathcal{M}_{d_3}]$ is $0.27 \times 0.04 = 0.0108$. Therefore, we would prefer document d_1 to documents d_2 and d_3 given the query.

5 Summary

We have outlined very briefly how to estimate the parameters (vector weights) for implementing the SPUD retrieval model. Although the model is a bag-of-words approach, it models the dependencies between recurrences of *word types* in a probabilistic framework. Given that most of the features used in the model are available directly from inverted indexes, there is little reason why the SPUD approach should not been seen as a viable superior approach to that of the standard multinomial language model.

References

[1] Ronan Cummins, Jiaul H. Paik, and Yuanhua Lv. A pólya urn document language model for improved information retrieval. *CoRR*, abs/1502.00804, 2015.

[2] Charles Elkan. Clustering documents with an exponential-family approximation of the dirichlet compound multinomial distribution. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 289–296, New York, NY, USA, 2006. ACM.

[3] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[4] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and*

Development in Information Retrieval, SIGIR '94, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[5] Karen Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[6] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions of Information Systems*, 22:179–214, April 2004.