Introduction to Global Weight Models

Includes a brief introduction to

MySQL Implementation of the Vector Space Model

Abstract –This is Part 5 of a tutorial series on Term Vector Theory. Several global weight models are discussed and a brief introduction to MySQL implementation of the Vector Space Model presented.

Keywords: global weights, idf, idfp, pivoted normalization, mysql
Published: 10-27-2006; Updated: 05-30-2016

Note: This article is part of a legacy series that the author published circa 2006 at http://www.miislita.com, now a search engine site. It is now republished in pdf format here at http://www.minerazzi.com, with its content edited and updated. The original articles can be found referenced in online research publications on IR and elsewhere.

Introduction

In Parts 1, 2, 3, and 4 of this series on vector space models for Information Retrieval (IR) we described several term weight strategies (Garcia, 2016a; 2016b; 2016c; 2016d).

We now focus our attention to one of the most popular vector space applications: The MySQL Implementation of the Vector Space Model. Since global weights are instrumental to this implementation, we discuss these first.

Inverse Document Frequency (IDF)

In 1972, Spärck-Jones proposed the *IDF* measure,

$$IDF_i = log\left(\frac{D}{d_i}\right) \tag{1}$$

where D is the size of a collection of documents and d_i the number of documents mentioning index term i (Spärck-Jones, 1972; 1973; 2004). In (1) the base of the logarithms does not matter. We can derive (1) using probability arguments.

Let d_i be the number of documents from D that mention an index term i. Then $p_i = d_i/D$ is the probability that a document from D contains an index term i. To smoothly compare very large and small p_i values, the probability scale is compressed by taking logarithms; i.e. $log(p_i) = log(d_i/D)$ As logarithms are additive, then for any two terms $log(p_1p_2) = log(p_1) + log(p_2)$; i.e., index terms are assumed to be independent. Therefore, $p_1p_2 = (d_1/D)(d_2/D) = (d_1d_2)/D^2$. Thus, for n number of terms, $p_1 \ldots p_n = d_1 \ldots d_n/D^n$.

Since $D >> d_i$, $log(d_i/D) < 0$. To avoid negative values, the ratio inside the parentheses is inverted and the result taken for a global weight, $G_i = log(D/d_i)$, now called the *inverse document frequency (IDF)*.

Spärck-Jones argued that *IDF* was somehow related to the notion of specificity and exhaustivity. She defined specificity as the level of detail at which a given concept could be represented by an index term *i* (Spärck-Jones, 1972; 2004). By contrast, she defined exhaustivity as the number of topics or themes indexed for a document, or to the level of detail with which a given topic is treated. Indexing exhaustivity, she argued, could be broadly defined as the number of index terms present in a document (Spärck-Jones, 1973).

IDF estimates specificity. Robertson eventually reformulated *IDF* as a global weight in the absence of relevance information (Robertson, 2004; Spärck-Jones, Walker, & Robertson, 2000a; 2000b). This is a particular scenario treated within the Okapi framework.

Intuitively, *IDF* shows that an index term mentioned in many documents weighs less than one which occurs in a few documents. For instance, *a, and, in, is, of,* and *the* are low-*IDF* terms as they tend to appear in many documents. These terms are not specific to a document and cannot be used to discriminate between documents. Conversely, rare and infrequently used index terms are high-IDF terms and can be used to discriminate between documents.

Eventually, Salton and co-workers (Salton & Yang, 1973; Salton, Wong, & Yang, 1975; Salton, 1983; Salton & Buckley, 1987) combined *IDF* scores, as global weights, with term frequencies, as local weights, and proposed what is nowadays known as the classic *Term Frequency-Inverse Document Frequency Model* (TF-IDF Model),

$$w_{i,j} = L_{i,j}G_i = f_{i,j}\,log\left(\frac{D}{d_i}\right) = f_{i,j}IDF_i \qquad (2)$$

IDF Probabilistic (IDFP)

If instead of considering number of documents we consider the odds of finding documents mentioning a given index term, a new global weight model can be proposed.

If $p_i = d_i/D$ is the probability that a document from D contains an index term i, then $1 - p_i$ is the probability of not containing said term. Therefore, the odd ratio $\frac{1-p_i}{p_i}$ is equal to $\frac{D-d_i}{d_i}$ and a probabilistic weight, that we shall call the *IDF* probabilistic, can be computed as

$$G_i = IDFP_i = log\left(\frac{1-p_i}{p_i}\right) = log\left(\frac{D-d_i}{d_i}\right) \quad (3)$$

Figure 1 depicts a curve of global weights for several values of p_i.

d_i	D	p_i	G_i
10	100	0.1	0.95
20	100	0.2	0.60
30	100	0.3	0.37
40	100	0.4	0.18
50	100	0.5	0.00
60	100	0.6	-0.18
70	100	0.7	-0.37
80	100	0.8	-0.60
90	100	0.9	-0.95

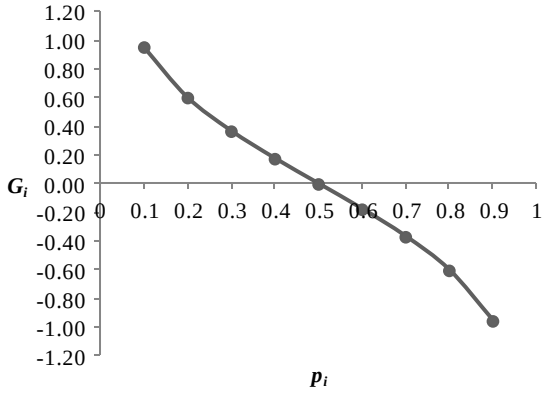

Figure 1. Graphical representation of the IDFP Model.

In the figure we used decimal logs. True that we could have used natural logs, but this is not a real issue as in this case the base of the logarithms does not really matter. Figure 1 shows that

- if less than 50% of the collection mentions index term i, $D/2 > d_i$ and $G_i > 0$.
- if 50% of the collection mentions index term i, $D/2 = d_i$ and $G_i = 0$.
- if more than 50% of the collection mentions index term i, $D/2 < d_i$ and $G_i < 0$.

Thus, the model assigns a weight of zero to terms mentioned in 50% of a collection and negative weights if they appear in more than 50% of a collection. Since terms with negative weights can introduce retrieval complications, these are frequently rescored as having zero weights, effectively behaving as redundant terms and stop words. These types of terms return too many results and are not useful for conducting searches. For instance, searching a video collection with [video] as the query is not an effective way of discriminating between search results.

Entropy-based Weights

An entropy function, *H*, is one of the form $P_i log_{base}\, P_i$ where *P* is a probability and *base* the base of a logarithm

Let $P_i = \frac{f_{i,j}}{F_i}$ where $f_{i,j}$ is the frequency of term *i* in document *j* and F_i is the frequency of term *i* in the entire collection. Then the entropy H_i of a term in a given document can be defined as

$$H_i \begin{cases} \frac{f_{i,j}}{F_i} \log\left(\frac{f_{i,j}}{F_i}\right) if\ f_{i,j} > 0 \\ 0\ if\ f_{i,j} = 0 \end{cases} \tag{4}$$

An entropy model (ENPY) can then be proposed by summing all entropies, $\sum H_i$, and log-normalizing them over the entire collection (Chisholm & Kolda, 1999)

$$G_i = 1 + \frac{\sum H_i}{\log D} = 1 + \frac{\sum\left(\frac{f_{i,j}}{F_i} \log\left(\frac{f_{i,j}}{F_i}\right)\right)}{\log D} \tag{5}$$

The 1 in (5) is added to force the scale of global weights to start at zero. For instance, if for a given term, $f_{i,j}$ is the same across all documents, the sum of all entropies equals –log*D*. Dividing by log*D* gives –1, and adding 1 gives $G_i = 0$ for said term.

By contrast, if a term is present once in just one document of the collection, $G_i = 1$. That term is given a global weight of zero. Accordingly, any other combination of term frequencies will yield a global weight somewhere between 0 and 1. Consequently, the model assigns higher weight to terms that appear fewer times in a small number of documents.

MySQL Implementation of Vector Space Models

MySQL (owned by the Oracle Corporation) is the world's most popular open source database. Their MySQL Internals Manual is an excellent source of information for developers interested in implementing a solid information retrieval solution.

Section 10.7 of the manual (MySQL, 2016), describes an interesting vector space implementation. The manual uses a nomenclature that, although different, is equivalent to the one used in this tutorial series. In the next section we discuss how they define local, global, and normalization weights.

MySQL Local, Global, and Normalization Weights

A typical MySQL vector space implementation uses database tables with N number of rows. Each row corresponds to a document, where

- $L_{i,j}$ = (log(dtf)+1)/sumdtf; i.e., a local weight based on logarithmic term counts
- N_j = U/(1+0.0115*U); i.e., a normalization weight based on pivoting normalization.
- G_i = log((N-nf)/nf); i.e., a global weight based on probabilistic IDF

where in MySQL implementation "log" means the log base *e* or natural logarithm (*ln*), and where

- dtf = number of times the term appears in the document
- sumdtf = sum of (log(dtf)+1)'s for all terms in the same document
- U = number of unique terms in the document
- N = total number of documents
- nf = number of documents that contain the term

After some rearrangements, the weight of a term is given by

$$w = (\log(dtf)+1)/sumdtf * U/(1+0.0115*U) * \log((N-nf)/nf) \quad (6)$$

which is the expression described in the developer's manual in section 10.7 (*Full-text Search*).

MySQL Local Weights

As mentioned, MySQL defines $L_{i,j}$ as (log(dtf)+1)/sumdtf, where sumdtf is the sum of (log(dtf)+1)'s for all terms in the same document and where stopwords are ignored. This is a fair alternative to the standard BNRY, FREQ, LOGA, LOGN, and ATF1 local weight models.

To illustrate, suppose that the title of a document is *MySQL Tutorial* and its content consists of the phrase *DBMS stands for DataBase*. Ignoring the *for* term, as it is a stopword, it must follows that

- unique index terms = 5; i.e., *mysql, tutorial, dbms, stands, database*
- (log(dtf)+1) = (log(1)+1) = 1 for each term
- sumdtf = 1 + 1 + 1 + 1 + 1 = 5
- $L_{i,j}$ = 1/5 = 0.2 for each term

Using the nomenclature of this tutorial series, this is the same as writing

$$L_{i,j} \begin{cases} \frac{1+\log(f_{i,j})}{\Sigma(1+\log(f_{i,j}))} \; if \; f_{i,j} > 0 \\ 0 \; if \; f_{i,j} = 0 \end{cases} \tag{7}$$

Essentially (7) is the LOGA model described in Part 4 of this tutorial series, with its scale normalized by summing all LOGA weights (Garcia, 2016d).

MySQL Normalization Weights

Document vectors are typically normalized by converting them into unit vectors and then computing the cosine similarity between these. In the classic IR literature this is called *cosine normalization* (COSN).

As noted by Chisholm & Kolda (1999), one limitation of COSN is that longer documents are given smaller individual term weights, so smaller documents are favored over longer ones in retrieval. This occurs because the L_2*-norm* (length of the vectors) used to convert raw vectors into unit vectors increases with increasing number of non-zero term weight values (Lee, 2010).

Pivoted Unique Normalization (PUQN) tries to overcome this by dealing with discrepancies between the probability that a document is relevant and the probability that the document will be retrieved. The point at which the resultant precision and recall curves intersect is the pivot. The technique can improve the quality of the retrieved documents.

Documents on the left side of the pivot generally have a higher probability of being retrieved than they have of being relevant. On the other hand, documents on the right side of the pivot tend to have a higher probability of being relevant than they have of being retrieved. The normalization factor can now be pivoted at the pivot and adjusted so that N_j can be increased or decreased to better match the probabilities of relevance and retrieval (Singhal, Buckley, & Mitra, 1996a; Singhal, Salton, Mitra & Buckley, 1996a; Singhal, Salton & Buckley, 1996c).

In the MySQL implementation, PUQN is defined as N_j = U/(1+0.0115*U), where U is the length of a document, defined as the number of unique terms. The 0.0115 quantity is a user-defined pivot value. See PIVOT_VAL in the MySQL source code header file myisam/ftdefs.h).

If U is shorter than the average length, computed over the entire collection of documents, the weight of the document increases. If it is equal to the average length then its weight stays the same, and if it is longer than the average length then its weight decreases. Thus for the above example,

- U = 5; i.e., *mysql, tutorial, dbms, stands, database*
- N_j = U/(1+0.0115*U) = 5/(1+0.0115*5) = 4.7281

MySQL Global Weights

MySQL defines two global weights score (GWS): G_i = log((N-nf)/nf) and G_i = log(N/nf) where, again, these are natural logs. So, if in the current example the document is part of a collection consisting of N = 6 documents and only two documents contain the word *"tutorial"*, the global weight of this term is

- if using *IDFP*, G_i = log((N-nf)/nf) = log((6-2)/2) = 0.6931
- if using *IDF*, G_i = log(N/nf) = log(6/2) = 1.0986

Evidently, these are just the *IDFP* and *IDF* global weight models mentioned early in this tutorial. Therefore, using *IDFP* and the naming convention of this tutorial series, (6) can be summarized as

$$w_{i,j} = \left(\frac{1+\log(f_{i,j})}{\sum(1+\log(f_{i,j}))}\right)\left(\frac{\mathrm{U}}{1+0.0115*\mathrm{U}}\right) log\left(\frac{D-d_i}{d_i}\right) \quad (8)$$

Thus, in the example the term weight of *tutorial* is $w_{i,j}$ = 0.20*4.7281*0.6931 = 0.6554. For routine work, if you are using *IDFP* and want to switch to *IDF*, MySQL allows you to easily do this. The manual suggests:

> "To go back to the old system, look in myisam/ftdefs.h for "#define GWS_IN_USE GWS_PROB" (i.e. global weights by probability) and change it to "#define GWS_IN_USE GWS_IDF" (i.e. global weights by inverse document frequency)".

To ease the switching between *IDFP* and *IDF* (or other global weight definitions), the $L_{i,j}*N_j$ product is stored in the column of the articles table generated from an index. Thus, in the example $L_{i,j}*N_j$ = 0.20*4.7281 = 0.9456. This product can then be used for whatever the global weight model implemented.

Finally, $w_{i,j}$ is multiplied by the number of times the term appears in the query, *qf*, and a rank or relevance score, R, computed

$$R = w_{i,j}*qf \quad (9)$$

In our example, if *tutorial* appears once in the query, R is $w_{i,j}*qf$ = 0.6554*1 = 0.6554. In (9), we are essentially applying (7) to document terms and the Term Count Model to query terms (Garcia, 2016b; 2016c).

Conclusion

We have shown that MySQL implementation of the Vector Space Model is based on computing a rank or relevance score, R, as the cross products between document and query weights. Term weights are computed using local, pivoted normalization, and global weights. Query weights are computed with the classic Term Count Model.

A big plus from MySQL is that allows developers to easily switch between *IDFP* and *IDF* global weights. It would be nice to have similar features for switching between other popular weighting models like global entropy weights or the several vector space models discussed throughout this tutorial series. This will make their implementation more versatile, robust, and a time-saving heaven for developers that are not well familiar with vector space theory. As it is now, these and similar auxiliary weighting strategies must be coded separately.

Exercises

1. A collection consists of five documents. A term X is mentioned only in 1 document of this collection and three times. Calculate its global weight using the *IDF*, *IDFP*, and ENPY models.
2. To convince yourself that when computing *IDF* and *IDFP* weights the base of the logarithms does not matter, reproduce Figure 1 using base *e* and base 2 logarithms.

References

Chisholm, E. and Kolda, T. G. (1999). New Term Weighting Formulas for the Vector Space Method in Information Retrieval. Oak Ridge National Laboratory. Retrieved from http://www.sandia.gov/~tgkolda/pubs/pubfiles/ornl-tm-13756.pdf

Garcia, E. (2016a). Term Vector Theory and Keyword Weights. Retrieved from http://www.minerazzi.com/tutorials/term-vector-1.pdf

Garcia, E. (2016b). The Binary and Term Count Models. Retrieved from http://www.minerazzi.com/tutorials/term-vector-2.pdf

Garcia, E. (2016c). The Classic TF-IDF Vector Space Model. Retrieved from http://www.minerazzi.com/tutorials/term-vector-3.pdf

Garcia, E. (2016d). An Introduction to Local Weight Models. Retrieved from http://www.minerazzi.com/tutorials/term-vector-4.pdf

Lee, L. (2010). Pivoted Document Length Normalization. CS 6740: Advanced Language Technologies, Lecture 3. Retrieved from https://www.cs.cornell.edu/courses/cs6740/2010sp/guides/lec03.pdf

MySQL (2016). MySQL Internals Manual. Retrieved from https://dev.mysql.com/doc/internals/en/full-text-search.html

Robertson, S. E. (2004). Understanding Inverse Document Frequency: On Theoretical Arguments for IDF. Reprinted from Journal of Documentation 60, 503-520. Retrieved from http://www.staff.city.ac.uk/~sb317/idfpapers/Robertson_idf_JDoc.pdf

Salton, G. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Salton, G. and Buckley, C. (1987). Term Weighting Approaches in Automatic Text Retrieval.87-881. Cornell University. Retrieved from https://ecommons.cornell.edu/bitstream/handle/1813/6721/87-881.pdf?sequence=1&isAllowed=y

Salton, G., Wong, A., and Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. Communications of the ACM 18 (11): 613. Retrieved from http://elib.ict.nsc.ru/jspui/bitstream/ICT/1230/1/soltan_10.1.1.107.7453.pdf
see also http://www.bibsonomy.org/bibtex/10a4c67f15a4869634d8e5e39ba3e7113

Salton, G. and Yang, C. S. (1973). On the Specification of Term Values in Automatic Indexing. TR 73-173, Cornell University. Retrieved from https://ecommons.cornell.edu/bitstream/handle/1813/6016/73-173.pdf?sequence=1&isAllowed=y

Singhal, S., Buckley, C., and Mitra, M (1996a). ACM SIGIR'96, 21-29, 1996. Pivoted Document Length Normalization. Retrieved from http://singhal.info/pivoted-dln.pdf

Singhal, S., Salton, G., Mitra, M., and Buckley, C. (1996b). Document Length Normalization. Information Processing and Management, 32:5, 619-633, 1996. Retrieved from https://ecommons.cornell.edu/bitstream/handle/1813/7186/95-1529.pdf?sequence=1

Singhal, A., Salton, G., and Buckley, C. (1996c). Length Normalization in Degraded Text Collections. Fifth Annual Symposium on Document Analysis and Information Retrieval, 149-162, 1996. Retrieved from http://singhal.info/ocr-norm.pdf

Spärk-Jones, K. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval. Journal of Documentation, Vol 60, 5, 493-502. Retrieved from http://www.staff.city.ac.uk/~sb317/idfpapers/ksj_orig.pdf

Spärk-Jones, K. (1973). Does Indexing Exhaustivity Matter? Journal of the American Society for Information Science. September-October, 313-316. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/asi.4630240502/epdf?r3_referer=wol&tracking_action=preview_click&show_checkout=1&purchase_referrer=onlinelibrary.wiley.com&purchase_site_license=LICENSE_DENIED

Spärck-Jones, K. (2004). IDF term weighting and IR research lessons. Reprinted from Journal of Documentation 60, 5, 521-523. Retrieved from http://www.staff.city.ac.uk/~sb317/idfpapers/ksj_reply.pdf

Spärck-Jones, K., Walker, S., and Robertson, S.E. (2000a). A probabilistic model of information retrieval: development and comparative experiments. Information Processing and Management 36, Part 1 779-808. Retrieved from http://www.staff.city.ac.uk/~sb317/blockbuster/pmir-pt1-reprint.pdf

Spärck-Jones, K., Walker, S., and Robertson, S.E. (2000b). A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management* **36**, Part 2 809-840. Retrieved from http://www.staff.city.ac.uk/~sb317/blockbuster/pmir-pt2-reprint.pdf