Comparative Analysis of IDF Methods to Determine Word Relevance in Web Document

Jitendra Nath Singh[1] and Sanjay K. Dwivedi[2]

[1] Department of Computer Science, Babasaheb Bhimrao Ambedkar University Lucknow, Uttar Pradesh, 226025, India

[2] Department of Computer Science, Babasaheb Bhimrao Ambedkar University Lucknow, Uttar Pradesh, 226025, India

Abstract

Inverse document frequency (IDF) is one of the most useful and widely used concepts in information retrieval. When it is used in combination with the term frequency (TF), the result is a very effective term weighting scheme (TF-IDF) that has been applied in information retrieval to determine the weight of the terms. Terms with high TF-IDF values imply a strong relationship with the document they appear in. If that term appears in a query, the document can be of most interest to the user. Term frequency is computed as the number of occurrences of a term in a document whereas there are various methods for measuring the value of IDF; one of the most famous derivations follows from the Robertson-Spark Jones relevance weight. Besides the most famous method for computation of IDF, there are also various methods for computation of inverse document frequency that affects the relevance of a document. In this paper, we have discussed and compared different derivations of inverse document frequency to measure the weight of terms.

Keywords: *Information Retrieval, Term-Frequency, IDF, Vector space model.*

1. Introduction

Information retrieval systems are designed to help users to find quickly useful information on the web page and judge the relevancy of a web page based on term weight. On the basis of relevancy, we can evaluate the performance of search engines. In this regard different information retrieval models used in the evaluation of search engines, such as vector space model [1,2,3,4] and probabilistic model [5] helped much and became the baseline for their framework and algorithms. Boolean model [5], also known as “exact match” model is still being used by most of the online services.

Search engines are used to retrieve useful information on the web using different search criteria. Relevance of retrieved information depends on the several tips and tricks of searching like simple search, advance search, default search mode, exact phrase mode, word truncation, word stemming and boolean queries etc. The default search mode is simple search where we can simply type the keywords according to our requirements and on search result page we can further add or remove keywords to get more precisely at what we are looking for. This phrase search mode will match if and only if the given metadata field is exactly equal to the input pattern. The single quotes instruct the search engine to search for partial phrases. Unlike the exact phrase search, this mode allows for an extra text appearing before or after giving patterns. This is somewhat similar to the "phrase search mode”. Advance search mode, we can change the matching type from default word matching to phrase searching or regular matching. Another interesting searching mode besides the word and phrase searches are the regular expression search, introduced by slashes instead of quotes.

Inverse document frequency (IDF) [12] is one of the most important and widely used concepts in information retrieval. It was first introduced by Spark Jones in 1976 with the aim of improving retrieval system. The IDF is used in combination with the term- frequency (TF). The result is a very effective term-weighting scheme [14] that has been applied for information retrieval systems. In information retrieval, computations of term-frequency and inverse-document frequency have a great importance because they evaluate the importance of words in a document. We assign a weight for each term in a document, which depends on the number of occurrences of that term in the document. We can compute a score between a query terms and documents based on weight of terms in the document. The simplest method to assign the weight to be equal to the number of occurrences of the term in the documents, this weighting scheme is referred as term frequency. The main problem with the term-frequency approach is that it scales up frequent terms and scales

down rare terms. The basic intuition is that a term that occurs frequently in many documents is not a good discriminator, and really does not make a sense. The TF-IDF weight comes to solve this problem. It tells how important a term in a document, and that's why TF-IDF incorporates local and global parameters, because it takes into consideration not only the isolated term but also the term within the document collection. What TF-IDF then does to solve that problem is to scale down the frequent terms while scaling up the rare terms. Inverse document frequency (IDF) is a popular measure of a word's importance. It is defined as the logarithm of the ratio of total numbers of documents in a collection to the number of documents containing the given query terms (Sparks Jones, 1976) [15]. It tells that rare query terms in the documents have a higher inverse document frequency and most common query terms have a low inverse document frequency.

2. VSM in Classical IR Perspective

The vector space model (VSM) represents documents and queries as vectors in multidimensional space, whose terms are used as dimensions to build an index to represent the documents.

Each dimension corresponds to separate term. If a term occurs in the document, its value in the vector is non-zero. It is used in information retrieval, indexing and relevant ranking and can be successfully used in evaluation of web search engines. The vector space model procedure can be divided into three stages. The first stage is the document indexing where content bearing terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of documents relevant to the user. The last stage ranks the documents with respect to the query according to a similarity value. A common similarity measure known as cosine measure determines the angle between the document vector and the query vector. The angle between two vectors is considered as a measure of divergence between the vectors. The cosine angle is used to compute the numeric similarity between the document vector and the query vector when they are represented in V-dimensional Euclidian space where V is the size. The classical method of vector space model to compute cosine similarity [3] between a document vector Di and query Q is given by:

$$sim(Q, D_i) = \frac{\sum_{j=1}^{V} w_{Q,j} \times w_{i,j}}{\sqrt{\sum_{j=1}^{v} w_{Q,j}^{\ 2}} \times \sqrt{\sum_{j=1}^{v} w_{i,j}^{\ 2}}} \qquad (1)$$

Where $w_{i,j}$ is the weight of term j in the document i and $w_{Q,j}$ is the weight of term j in the query. The denominators in this equation, called the normalization factor, discard the effect of document length on document scores.

3. An Overview of TF-IDF

Inverse document frequency (IDF) [12] is one of the most important and widely used concepts in information retrieval. It is used in combination with the term-frequency (TF). The result is a very effective term-weighting scheme that has been applied for information retrieval systems. We combine the term-frequency and inverse document frequency to produce a composite weight of each term in a document i.e. TF-IDF scheme. The TF-IDF method computes the weight of terms [7, 14] using the following equation.

$$w_j = TF \times IDF \qquad (2)$$

Where Wj is the weight of term j in a document (or query), TF is the term frequency (number of occurrences of a query term in a document) and IDF is the inverse document frequency.

4. Derivation of IDF

After analyzing different approaches of weighting scheme of vector space model [8], it was concluded that TF-IDF method [14] is the most suitable weighting model for computation of weight. Further, the IDF has a vital role in computation of weight of terms in queries and documents using TF-IDF approach. If IDF value is high, the weight of the term in the documents will be high. It means the document is more favorable to the user interest. In 2000, Li and Y Shang [2, 17] presented that it is very difficult to compute inverse document frequency (IDF) because computation of the total number of documents and number of documents containing the query terms are not possible. So they set the value of inverse document frequency constant for the search term. But this is feasible only when all the terms used in performance evaluations are common technical terms that appear approximately the same number of times.

If we make IDF constant, then the computation of term weight only depends on term frequency which is term-count model. IDF depends on availability of terms in the documents and also shows the importance of words and may be changed in different situation. By going through different literatures [7, 8, 11, 14, 6, 9, 10], we found various derivations of IDF computations. Among them, we use four most commonly used IDF methods for our analysis.

4.1 Method I

The simple and most commonly form of method for computation of IDF [7, 8, 11, 14] is given by:

$$idf = \log\left(\frac{D}{df_j}\right) \qquad (3)$$

D is the number of documents in the document collection and df_j a number of documents containing the query term. This is the most commonly cited forms of IDF; some other forms are shown below.

4.2 Method II

The IDF is computed [6] given by:

$$idf = \log\left(\frac{D+1}{df_j}\right) \qquad (4)$$

4.3 Method III

Another method for computation of IDF [9] given by:

$$idf = \log\left(\frac{D}{df_j}+1\right) \qquad (5)$$

4.4 Method IV

Another method for computation IDF [10] is squared inverse document frequency given by:

$$idf = \log\left(\frac{D}{df_j}\right)^2 \qquad (6)$$

Hence IDF will be computed based on Eq. (3), Eq. (4), Eq. (5), and Eq. (6) .These equations of the IDF can have been used in the computation of the weight of the terms in the document.

5. Experiments

In our analysis, we applied different methods of IDF mentioned above (Eq. 3,4,5 & 6) for computing the weight of terms found in retrieved documents (for each query) using TF-IDF approach. These experiments were based on an accepted number of TREC pattern queries involving Google search
These queries contain 2, 3 terms. A query set containing 50 queries, query Id from 1 to 50 given in Table 1.
There are various search modes but we have applied keyword based or default search mode. We propose a simple approach to compute the weight of query terms in the document based on these IDF methods as given below

- Prepare a set of sample queries (TREC pattern)
- Submit each query on target search engine (by considering only top 10 links).
- Follow each web page and compute the weight of the terms in the document (if those terms also appear in the query) using following steps.

  • Compute Term frequency (TF) i.e. number of occurrences of a query term in a document, thereafter values of the IDF based on these four methods.
  • Based on these term frequencies and IDF, we compute weight of each term in the documents, if those terms also found in the query using Eq. (2) (i.e. TF-IDF method).

The computed weights for 20 queries are given in Table 2 (Appendix A).

6. Discussion

Based on the experiments, we have made certain observations. Relevancy of documents depends on TF-IDF weight of the terms in the document and IDF plays important role in the computation of the weight of the terms. If weight of a term in the document is computed high using TF-IDF approach and the term is also available in the query, the document has more relevance for the given query term. Using the queries given in Table 1, we computed TF-IDF weight of query terms that are found in the retrieved documents using various methods of IDF as given in Table 2(Appendix A).

Table 1: Selected Query (on TRECK Pattern)

1: iodine in blood	11: job safety analysis	21: food services	31:new orleans	41:radon inspector
2: student jobs	12: adobe Indian houses	22:wright brothers	32:optional form 306	42: local civil rule 83.3
3:weight of mail	13: arizona game and fish	23:school bus safety	33:chester an arthur	43:storium 90
4: global warming	14:feta cheese preservatives	24: nuclear commission	34:action plan	44:symptoms of heart
5: loan proposal	15: credit report	25:listeria infection	35:attorney for senior	45:weather strip
6: surface area evaporation	16:quit smoking	26:signature of first ladies	36:eta form 9089 dl	46:check my status
7: corn price	17:black history	27:online coloring books	37:family education	47:civil right movement
8: energy from	18: computer programming	28:capital hill massacre	38:unique rare coins	48:credit report
9:weather radar	19:sore throat	29:earthquake in california	39:diarrhea pregnancy	49:internet phone service
10: march health awareness	20:survey maps	30:gangster disciples	40: hand washing gel	50: brooks brothers clearance

The weights of certain queries as computed by method I and method IV have the value zero when all query terms are found in all the documents (for top ten retrieved documents as we considered only top ten documents for each query). Though the weight reflects that these documents are of no use to the query, actually they can be of most interest to the user. Clearly, the IDF computations used in these two methods are not very suitable to measure the relevance of a document in such situations. TF-IDF weight computed by method II is better than method I but far less than method III (Graph 1). The IDF computation of Method IV is similar to that of method I that is if method I provide zero weight to a query the same weight is provided by method IV also; otherwise it gives better values than method I. In comparison of various methods of IDF, we observed that methods I and IV often fail to compute the IDF value because query terms present in all documents whereas the other two methods do compute the weights in such circumstances also. Therefore the methods II and III are better suited for IDF computation. Hence the experimental results of the comparison of four methods of IDF using keyword based search suggest that the IDF value computed by method III gives better TF-IDF weight of terms compared to other three methods. So this method may be used in computation of similarity values using Eq. (1) in order to rank documents which depends on similarity value.

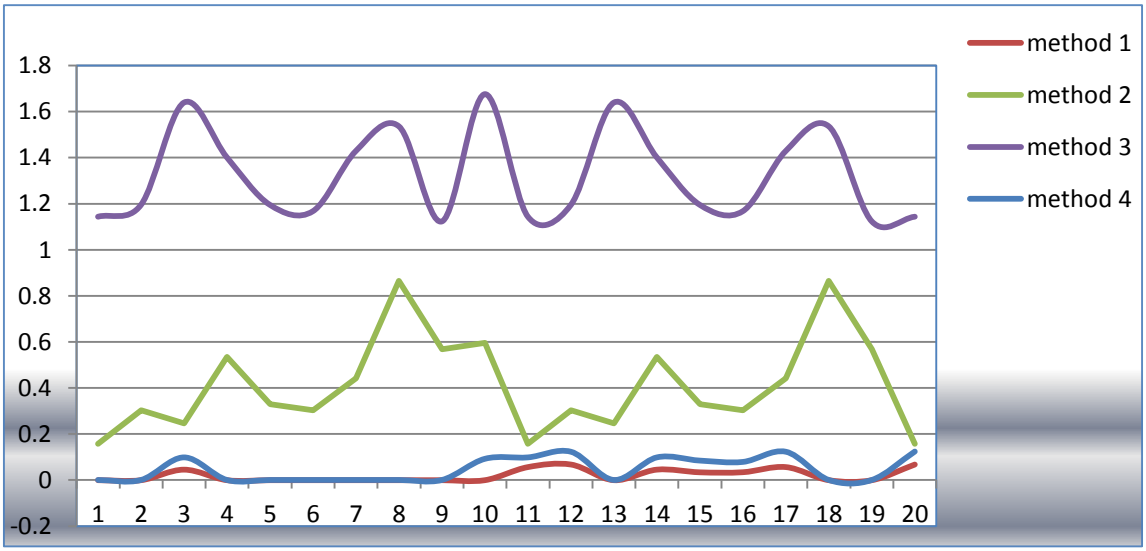

Fig 1 Comparison of IDF Methods based on Top Ten Hits and Average Score per Page.

7. Conclusion

In this paper, we have discussed the relevancy of a web page based on weight of terms computed using TF-IDF method. The computation of term frequency is not a big issue. We therefore focused only on inverse document frequency in computation of weight of terms. We compared four methods of IDF found in literature using a set of TREC based queries. After computing the value of the IDF using keyword based search, we concluded that the IDF value computed by method III gives better TF-IDF weight of terms compared to other methods. This is also true in the situations wherein other methods fail to give good weights. It was also concluded that the relevancy of web page depends on weight of terms computed by TF-IDF methods. A query terms with high weight imply a strong relationship with the documents they appear in. Based on these TF-IDF values, the cosine function computes the similarity values between queries and documents for ranking the documents. The analysis done in this paper may indeed be utilized to implement the best formula of IDF according to the search scenario in order for better evaluation of search engines.

References

[1] Shalton, G; Wong A, Yang C.S, "A vector space Model for automatic indexing", Communications of the ACM, Volume 18, No. 11, 1975.

[2] Longzhuang Li, Yi Shang, "A new statistical method for performance evaluation of search engines", ICTAI: 2000.

[3] Dik L. Lee, "Document ranking and the Vector-space model. Software", IEEE, Vol.14, No.2, pp. 67-75, Mar-April.1997.

[4] Longzhuang Li, Yi Shang, "A new method for automatic performance comparison of search engines", World Wide Web: 2000.

[5] Djoerd Hiemstra, "Information retrieval models", In Information Retrieval: Searching in the 21st Century. Wiley. 2009.

[6] Chris Buckley, "The importance of proper weighting methods", In M. Bates, editor, Human Language Technology. Morgan Kaufman: 1993.

[7] Gerald Salton & Chris Buckley, "Term weighting approaches in automatic text retrieval", Information Processing and Management, 24(5): No. 5. 1988.

[8] Jitendra Nath Singh & Sanjay Kumar Dwivedi, "Analysis of Vector Space Model in Information Retrieval", Proceedings (IJCA) on National Conference on Communication Technologies & its impact on Next Generation Computing 2012 CTNGC (2): pp.14-18, 2012. .

[9] Y Jung, H Park, D Du, "An effective Term- weighting scheme for Information Retrieval", Technical Report TR00-008 Department of Computer Science and Engineering, University of Minnesota, 2000.

[10] Nicola Polettini, "The Vector Space Model in Information Retrieval-Term Weighting Problem", 2004.

[11] Stephen, "Understanding Inverse Document Frequency: on theoretical arguments of IDF", Journal of documentation Vol 60, No. 5, pp 503-520.

[12] Kishore Papineni, "Why inverse document frequency", Proceedings of the North American Association for Computational Linguistics, NAACL, pp. 25–32, 2001..

[13] S .Takao, J. Ogata, Y. Ariki, "Study on New Term Weighting Method and New Vector space model based on Word Space in Spoken Document Retrieval", RIAO00, Volume I, pp. 116-131, 2000-04.

[14] J. Ramos, "Using TF-IDF to determine word relevance in document queries", In First International Conference on Machine Learning, New Brunswick: NJ, USA, 2003..

[15] [K. Spark. Jones, "A statistical interpretation of term specificity and its application in retrieval", Journal of Documentation, Vol 28, No.1, pp. 11–21, 1972. .

[16] A. Singhal, C. Buckley, and M. Mitra, "Pivoted document length normalization", In Proceedings of the 19th Annual International ACM SIGIR Conference on Research..

[17] Yi Shang Longzhuang Li, "Precision Evaluation of Search Engines", World Wide Web pp.159-173, 2002. .

Sanjay K. Dwivedi Associate professor, Department of computer science at Babasaheb Bhimrao Ambedkar University, Lucknow 226025 (U.P.) India. His research interest is in Artificial Intelligence, web Mining, NLP and sense disambiguation etc. He has 16 years of experience of teaching and research and has handled/involved in some government funded research projects. He has published a large number of research papers in reputed International journals and conferences.

Jitendra Nath Singh Research Scholar in Department of computer Science at Babasaheb Bhimrao Ambedkar University, Lucknow - 226025 (U.P.) India. His research interest is search engines and its performance evaluation, and web technology.

Appendix A

Table 2: Weight of Terms Using Four Methods of IDF for 20 Queries.

QUERY ID	IDF METHODS	WEIGHT OF QUERY TERMS PRESENT IN DOCUMENTS									
		D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
1	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	0.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	0.3592	.2209	0.4975	0.3593	0.2209	0.2209	.2209	0.3592	0.2209	0.3592
	III	1.306	0.9541	1.658	1.306	0.9541	0.9541	0.9541	0.9582	0.9541	0.9541
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	I	0.0457	0.0239	0.0239	0.0779	0.0688	0.0438	0.0184	0.0488	0.0456	0.0308
	II	0.1791	0.1239	0.1239	0.1779	0.1688	0.1138	0.1184	0.1688	0.1123	0.1708
	III	1.375	1.987	0.9876	0.9870	1.234	1.5670	0.9876	1.234	1.345	0.9967
	IV	0.0915	0.0449	0.0490	0.1190	0.1160	.0839	0.0340	0.0960	0.0987	0.0645
4	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	0.0221	0.0191	0.0221	0.0181	0.0234	0.0194	0.02821	0.0196	0.0198	0.0231
	III	1.234	0.9780	1.234	0.9987	1.234	0.8870	0.7890	0.9987	.09876	.0678
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	0.0351	0.0468	0.0457	0.0210	0.0551	0.0743	0.0468	0.03468	0.0382	0.0675
	III	0.1234	0.2334	0.1123	0.1234	0.1123	0.1211	0.1134	0.1234	0.1212	0.1123
	IV	0.0251	0.0165	0.0025	0.0020	0.0012	0.016	0.0160	0.0123	0.0090	0.0
6	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	0.0451	0.0568	0.0557	0.0310	0.0651	0.0843	0.0668	0.04468	0.0482	0.0575
	III	0.2234	0.2534	0.2123	0.2234	0.2123	0.2211	0.2134	0.2234	0.2212	0.2123
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	0.0251	0.0368	0.0357	0.0310	0.0451	0.0743	0.0568	0.0346	0.0482	0.0575
	III	0.1434	0.2434	0.1423	0.1434	0.1323	0.1211	0.1234	0.1234	0.1212	0.1223
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	0.0356	0.0469	0.0459	0.0219	0.0558	0.0748	0.0469	0.0346	0.0385	0.0575
	III	0.1434	0.2134	0.1423	0.2234	0.2123	0.2211	0.2134	0.2234	0.3212	0.2123
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	0.0356	0.0469	0.0459	0.0219	0.0558	0.0748	0.0469	0.0346	0.0385	0.0575
	III	0.1234	0.2334	0.1123	0.1234	0.1123	0.1211	0.1134	0.1234	0.1212	0.1123
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	0.0451	0.0568	0.0657	0.0410	0.0851	0.0943	0.0968	0.0546	0.0482	0.0775
	III	0.2234	0.2834	0.2123	0.1934	0.1923	0.1911	0.2134	0.2234	0.2212	0.2123
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

12	I	0.03461	0.0456	0.0452	0.0531	0.0456	0.0345	0.0543	0.0345	0.0675	0.0456
	II	.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0657	0.0965	0.0987	0.0991	0.0987	0.0786	0.0999	0.0765	0.1234	0.0987
13	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	I	0.03461	0.0456	0.0452	0.0531	0.0456	0.0345	0.0543	0.0345	0.0675	0.0456
	II	.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0657	0.0965	0.0987	0.0991	0.0987	0.0786	0.0999	0.0765	0.1234	0.0987
15	I	0.03461	0.0456	0.0452	0.0531	0.0456	0.0345	0.0543	0.0345	0.0675	0.0456
	II	.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0657	0.0965	0.0987	0.0991	0.0987	0.0786	0.0999	0.0765	0.1234	0.0987
16	I	0.03461	0.0456	0.0452	0.0531	0.0456	0.0345	0.0543	0.0345	0.0675	0.0456
	II	.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0657	0.0965	0.0987	0.0991	0.0987	0.0786	0.0999	0.0765	0.1234	0.0987
17	I	0.03461	0.0456	0.0452	0.0531	0.0456	0.0345	0.0543	0.0345	0.0675	0.0456
	II	.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0657	0.0965	0.0987	0.0991	0.0987	0.0786	0.0999	0.0765	0.1234	0.0987
18	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	II	.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	I	0.0657	0.0965	0.0987	0.0991	0.0987	0.0786	0.0999	0.0765	0.1234	0.0987
	II	.1652	0.1239	.1652	0.2065	.1239	0.1239	0.1239	0.2478	0.1239	0.1652
	III	1.204	0.903	1.204	1.505	0.903	0.903	0.903	1.806	0.903	1.204
	IV	0.0657	0.0965	0.0987	0.0991	0.0987	0.0786	0.0999	0.0765	0.1234	0.0987