A COMPARISON OF TWO METHODS FOR SOFT BOOLEAN OPERATOR INTERPRETATION IN INFORMATION RETRIEVAL

BY

EDWARD A. FOX
SHARAT SHARAN

TR-86-1

JANUARY 1986

A Comparison of Two Methods for Soft Boolean Operator Interpretation in Information Retrieval†

Edward A. Fox
Sharat Sharan

Department of Computer Science
Virginia Tech
Blacksburg VA 24061

ABSTRACT

Information retrieval systems generally are given Boolean logic queries by users or search intermediaries, in order that an efficient and effective search for relevant documents can be automatically carried out. Previous work with an extended interpretation of Boolean queries has shown that a dramatic improvement in search effectiveness results. Using the L_p–norm to compute distance from the ideal points in a multi–dimensional space of truth values leads to best results when p–values are on the order of 1 to 4.

Other schemes besides the "p–norm" approach have been proposed in recent years. This paper describes experimental studies aimed at evaluating one family of such methods. In particular, a parameterized fuzzy–logic approach is contrasted with the p–norm interpretation. Regression analysis supports expected results of parameter settings and gives further insight into why the p–norm scheme is superior.

CR Categories and Subject Descriptors H.3.1 **[Information Storage and Retrieval]**: Content Analysis and Indexing; H.3.3 **Information Storage and Retrieval]**: Information Search and Retrieval.

General Terms: Experimentation, Measurement

Additional Keywords and Phrases: Boolean retrieval, fuzzy sets, p–norm retrieval

---

† This work has been funded in part by the National Science Foundation under grant IST-8418877 and by the Virginia Center for Innovative Technology under grant INF-85-016

INTRODUCTION

Information retrieval systems may be given Boolean logic queries describing desired searches for relevant documents. Interpretation of these queries can be very efficient, since AND may be implemented by intersecting lists of document identifiers, and OR implemented by list union. The result is often very precise[1], particularly when a number of conjuncts are present. However, recall[2] is often rather low, especially if free–text collections are employed [2]. It is well known that recall and precision are inversely related [8], so it is clear that when other queries are used which give higher recall, then lower precision would be expected. Approaches that lead to higher precision than some base case for each given recall level, are said to be more effective.

The p–norm method was proposed so that Boolean logic queries could be used to carry out more effective searches [7]. Extensive experimentation has demonstrated the benefits of this approach [3]. Since that time, however, other extended Boolean interpretations have been described but not thoroughly tested [6], [10].

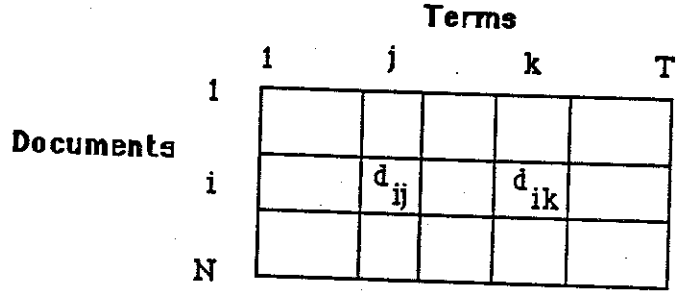
To set the context for this report, the Boolean and p–norm approaches to information retrieval are briefly outlined in the next section. For further details on Boolean searching the reader is referred to [5]. Similarly, the reader may wish to study [7] for more details on the p–norm interpretation of queries.

The subsequent section describes the p–norm method and reviews the experimental evidence regarding its performance. The following section describes the soft–Boolean interpretation suggested by Paice [6]. Next, the experimental design for carrying out a comparison is presented. Results are then given, and regression and other tests described. Preliminary findings are discussed and possible future subsequent studies are listed. Finally, this report is summarized and key conclusions explained.

BOOLEAN RETRIEVAL

Given a query Q and a collection C of N documents, it is desirable to select D_Q^R, the set of documents relevant to query Q. When N is very large, computers are needed to automate the search process. Documents are *indexed* by T terms[3] and queries are built up using some of those terms, so that automatic matching is possible. The collection representation may be viewed as a sparse matrix as shown in Figure 1.

Figure 1. Matrix Representation for Collection C



[1] Precision is the ratio of number of relevant retrieved to the number retrieved.
[2] Recall is the ratio of the number of relevant retrieved to the number relevant.
[3] A term may be an author's name, a word (or word stem) in the document text, a subject descriptor assigned by the indexer, etc.

A document D_i is actually a vector of length T, $\{d_{i1}, d_{i2}, \ldots, d_{iT}\}$ where d_{ij} is 1 when the jth term is assigned to the ith document, and zero elsewhere.

Boolean queries can easily be understood in the context of this representation. A query

$$Q_{AND} = T_j \text{ AND } T_k$$

retrieves

$$D^{Bool}_{Q_{AND}} = \left\{ d_i \mid d_{ij} = 1 \bigwedge d_{ik} = 1 \right\}$$

so the Boolean interpretation for this AND query is the logical AND of vectors T_j and T_k, giving an intersection set.

Similarly, query

$$Q^{Bool}_{OR} = T_j \text{ OR } T_k$$

retrieves

$$D^{Bool}_{Q_{OR}} = \left\{ d_i \mid d_{ij} = 1 \bigvee d_{ik} = 1 \right\}$$

so the Boolean interpretation for the OR query is the logical OR of vectors T_j and T_k, giving a union set. When N is very large, it is useful for improved precision to use AND in queries since often

$$| D^{Bool}_{Q_{AND}} | \ll | T_j |, | T_k | .$$

OR queries are especially important for improved recall, especially when terms T_j and T_k are nearly synonymous, i.e., can be viewed as "searchonyms" [1], since

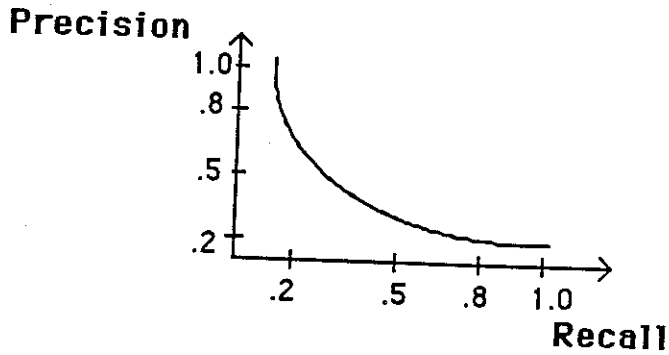
$$| T_i |, | T_j | \leq D^{Bool}_{Q_{OR}} \leq | T_i | + | T_j |$$

The construction of good Boolean queries to find relevant articles in large online collections is a difficult process that is often undertaken by information specialists trained to aid the actual end users. A searcher must:

1. understand a user's information need
2. select terms relating to that need which appear in the collection:
   a. guess at words the author of a relevant article might have used in the title, abstract, or perhaps in the full text if such is searchable
   b. guess at descriptors assigned by indexers to an article.
3. decide on a strategy for constructing a Boolean query from those terms, considering:
   a. which terms should be ORed to broaden the ability of a single term to describe some concept
   b. which terms should be ANDed to narrow or restrict results to include a combination of concepts
   c. how OR and AND clauses should be combined to describe higher level constructs.
4. relating the conceptual structure of a question to the statistics of co–occurence of search terms so a Boolean query can reflect both considerations.

Unfortunately, it is rare that a Boolean query comes close to retrieving all and only those articles relevant to the user's information need. Resulting queries are classified as narrow vs. broad, depending on whether a few or many items are expected. Experimental studies indicated that as recall increases, precision decreases, as shown in Figure 2.

Figure 2. Recall – Precision Inverse Relationship



Due to these limitations, and based on experience with Boolean queries, a number of questions have been raised:

1. Can queries be devised that give both very high recall and very high precision?
2. Is there a way to interpret Boolean queries so that such improved effectiveness will result?
3. Can computers be used to help in these two areas?

Drawing on insights from fuzzy set theory, the p–norm interpretation of Boolean queries has been developed to address these issues.

P–NORM RETRIEVAL

Key assumptions of the p–norm approach are:

1. Indexing is a fuzzy process so one should have $0 \leq d_{ij} \leq 1$.
2. "Strict" interpretation of AND and OR is inappropriate since linguistic semantic relationships in effect do not really correspond to statistical reality for retrieval.
3. A query should define a fuzzy set so that documents can be presented to users in order of decreased probability of relevance.

The first point is handled by allowing more flexibility in the values of d_{ij}. Those values should reflect a membership function, eg. describe the degree to which the ith document should be indexed by the jth term. Several formulations have been proposed; a key consideration is that terms which occur often in a document are more likely to characterize it than terms that occur only a few times. Experiments with the p–norm scheme have shown that this technique for determining d_{ij} leads to better retrieval than when binary values are employed [7].

In this study, a real value is computed for the membership function in three steps. First, the ratio (*ratio1*) of the frequency of a term in the document to that of the most frequent term in that document is computed. The term–frequency factor (TF) is, for terms occurring in the document,

$$TF = \frac{(1 + ratio1)}{2}$$

and 0 otherwise. Next, the effect of collection wide statistics is considered. The term and inverse document frequency value (TFIDF) is thus also based on the inverse document frequency:

$$TFIDF = TF \times \log(\frac{N}{\text{term\_coll\_freq}})$$

Finally, that value is normalized to the range [0,1] by dividing each TFIDF value by the sum of the squares of such values.

Regarding the strictness of interpreting AND and OR, it is worthwhile to first consider some examples. Query 1 below,

Q1: information AND retrieval AND system AND evaluation AND method

is the conjunction of 5 terms which reflect concepts that a user might wish to see included in an article. However, it is likely that there will be relevant articles where 3 or 4 of the terms are present (eg. retrieval/evaluation/method) – such articles would be strictly excluded by the normal Boolean view of Q1. Similarly, in query 2 below,

Q2: metric OR measure OR evaluation OR measurement

it is more likely that articles with two or more of these four terms present (eg. evaluation and measure) would be more relevant than if only one term is found (eg. "in my evaluation ...").

Regarding a fuzzy set of documents being determined, it seems clear that a ranked set as shown in part A of Figure 3 is more useful than an unranked set as shown in Part B of that Figure. Here similarity is the term used to describe how closely a document satisfies the query; similarity defines the desired fuzzy set of documents as a characteristic function.

Figure 3. Retrieved Document Sets

A. FUZZY SET

Document	Similarity
ID001	1.0
ID742	.834
ID819	.632
⋮	⋮
ID253	.104
ID006	0.0
⋮	⋮
ID997	0.0

B. BOOLEAN

Document	Similarity
ID001	1.0
⋮	⋮
ID029	1.0
ID003	0.0
⋮	⋮
ID997	0.0

In the Boolean case, if the retrieved set is too large or too small then a new query must be constructed. For the same Boolean query, however, as long as there is some use of the AND

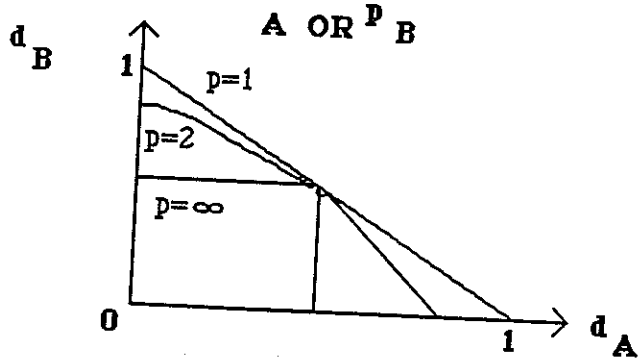
operator, the retrieved set for a soft Boolean evaluation will be larger, but the user need only look at those items with greatest similarity to the query.

The p–norm interpretation of AND and OR integrates all of these factors. First, d_{ij} is a real–valued measure, in range [0, 1], of the membership function for $term_j$ in connection with $document_i$. Second, for either an OR or an AND clause, a similarity value in range [0, 1] is defined measuring how well a given document satisfies that expression. By recursive application, the similarity of a query to a document is determined.

Next, a parameter p is introduced to allow variation in the strictness of interpretation of the AND and OR operators. When $p = \infty$, a strict interpretation is employed – essentially a fuzzy set theoretic computation. When $p = 1$, a loose interpretation is employed – essentially a vector inner–product or average calculation. Intermediate values give intermediate results. This can be best understood in the case of a single query, A OR B, by referring to Figure 4.

Figure 4. Equation and Contours for 2-term OR Clauses

$$SIM(A\ OR^P\ B,\ D) = \left\{ \frac{d_A^P + d_B^P}{2} \right\}^{\frac{1}{p}} = 2^{\frac{-1}{p}} \parallel D \parallel_p$$

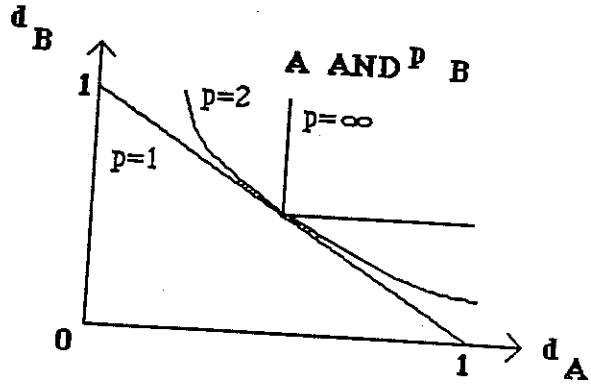


The two axes indicate the membership function values for a document in terms of term A (d_A) and term B (d_B). X, Y, Z are contours of equal similarity points in the d_A–d_B plane. They are defined by p–values of 1, 2, and ∞, respectively. That is, when $p = 1$, any combination of d_A-d_B values on line X will give the same similarity result for the given query.

For OR queries, similarity follows the intuition that one would like to be as far as possible from having no terms present. Thus, similarity is a normalized distance from the origin. AND queries are interpreted in terms of decreasing distance from the case of having all terms fully present, i.e. the 1 point. Figure 5 shows contours and equations.

Figure 5. Equation and Contours for 2-term AND Clauses

$$SIM(A\ AND^P\ B,\ D) = 1 - \left\{ \frac{(1-d_A)^p + (1-d_B)^p}{2} \right\}^{\frac{1}{p}} = 1 - 2^{\frac{-1}{p}} \parallel 1 - D \parallel_p$$



It should be noted that when p =∞, these equations simplify as shown below to familiar fuzzy set formula.

$$SIM(A\ OR^P\ B,\ D) = max(d_A, d_B)$$

$$SIM(A\ AND^P\ B,\ D) = min(d_A, d_B)$$

Furthermore, p–norm operators can be generalized to handle many terms in a clause instead of just two, and to handle user specified relative weights on each of the query clauses or terms [7]. To complete the definition of p–norm operators, the NOT case is given by

$$SIM(NOT\ expression, D) = 1 - SIM(expression, D)$$

Based on this scheme, significant improvements have been demostrated on a number of test collections. A further examination of the effect of p–values on retrieval performance is discussed below.

PAICE (Mixed *Min* and *Max* – MMM)

Another proposal for "softening" the strictness of Boolean logic expressions was made by Paice [6]. His scheme more closely follows fuzzy logic theory.

Zadeh's initial suggestion was to use *min* for AND and *max* for OR [11]. In retrieval situations, however, it may not be appropriate to only consider the worst term in an AND clause and the best term in an OR clause. Therefore, Paice suggested defining each operator as a linear combination of *min* and *max*. Intuitively, one would expect that this MMM (mixed min and max) scheme would be better than a strict interpretation, and that is what Paice found in some small-scale experiments.

Paice suggested defining

$$\text{A OR B OR C} = c_{OR,1} \times max(d_A, d_B, d_C) + c_{OR,2} \times min(d_A, d_B, d_C)$$

$$\text{A AND B AND C} = c_{AND,1} \times min(d_A, d_B, d_C) + c_{AND,2} \times max(d_A, d_B, d_C)$$

where usually $C_{OR,1} > C_{OR,2}$ and $C_{AND,1} > C_{AND,2}$ since OR should be more similar to *max* than to *min*, and since AND should be more similar to *min* than to *max*. For simplicity, he recommended setting, for both AND, OR

$$c_{AND,2} = 1 - c_{AND,1}$$

and

$$c_{OR,2} = 1 - c_{OR,1}$$

Paice did consider how to choose values for $c_{OR,1}$ (henceforth C_{OR}) and $c_{AND,1}$ (henceforth C_{AND}). He gave some brief explanation, but no firm guidelines. In the following section, an empirical study is described dealing with p-values for AND and OR clauses (specifically, how to select P_{AND}, P_{OR}) in p-norm queries, and comparing those results with MMM (Paice) coefficients for AND and OR (specifically, C_{AND} and C_{OR}).

EXPERIMENTAL DESIGN

A collection of 1460 documents on information science was chosen for initial experimentation [4]. There were 35 Boolean queries. Experts had decided which document is relevant to which question, so definitive recall and precision measures can be computed for each query. Furthermore, recall and precision can be averaged for a given query interpretation method, over the retrieved set for each query and over the set of all queries. Thus, a single "average precision" value can be determined for each experimental search method.

A recent study by Salton and Voorhees [9] examined a few of the possible p-norm combinations. This study was a follow up to conjectures and earlier comparisons described in [3]. They concluded that:

1) true synonyms ORed together could be connected with P_{OR} slightly higher than 1, but since real synonyms are rare, using low p-values for OR is best
2) higher values of P_{AND} are useful, especially when the terms connected by AND are somewhat independent (i.e., form a noun phrase)

In addition to these heuristics, the other hypotheses proposed at the start of the current investigation are:

1) low p-values are best for p-norm searches [based on previous results with several other collections.]
2) because of properties of the L_p family of norms, varying p in this range, eg. $1 \leq p \leq 4$, should have little absolute effect on effectiveness results
3) p-norm results will be better than MMM results [since more terms are considered in each p-norm clause than just the *min* and *max*]
4) fairly high values for C_{AND} and C_{OR} should give best results with the MMM scheme [since users do have wisdom in selecting AND and OR], but an overly strict interpretation is inappropriate

In order to test these hypotheses, a number of searches were made. For MMM tests, C_{AND} and C_{OR} were each varied from 0.0 to 1.0 in steps of 0.1, giving a total of 11 × 11 or 121 cases. For p–norm tests, P_{AND} and P_{OR} were varied from 1 to 4 in increments of 0.2, giving a total of 16 × 16 or 256 runs. Other key cases were tested such as having both p–values set to 5, 6, 7, 8, 9, 10, 15, ∞. In addition, some special exploratory runs were made later with other interesting combinations. A heuristic case, where each query operator had p–value set by the second author of this work, according to the rules suggested by Salton and Voorhees (see above), was also included. An example is the twelfth query, which in p–norm prefix form is

#and 4 (#or 2 (publication, printing, distribution),
#or 1 (methods, scientific, journals))

RESULTS

Table 1 shows the average precision values for p–norm runs with $1 \leq p \leq 4$. Table 2 summarizes the remaining p–norm runs. Table 3 shows average precision values for MMM runs. To help interpret these values, contour plots for p–norm and MMM scheme are shown in Figures 6 and 7, respectively.

It can be seen from Tables 1,2 or from Figure 6 that for this document and query collection, p–norm results depend primarily on the p–value for the OR operator and are little effected by the p-value on the AND operator. To highlight this fact, Figure 8 shows how average precision varies with the p-value for ORs, when p–value for AND is fixed. For contrast, Figure 9 shows how little average precision changes when p–value for AND is varied, for fixed OR p-value.

In similar vein, Figure 10 is for MMM when C_{OR} is varied and C_{AND} is fixed. Figure 11, which looks much the same, is for MMM when C_{AND} is varied and C_{OR} is held constant.

To summarize and compare the various schemes being considered, Table 4 gives the average precision for selected cases.

INTERPRETATION

The overall trend of results, shown in Table 4, is clear. Standard Boolean methods give lowest average precision. Next comes the p–norm scheme, with p–values set to ∞. This is very close to running MMM with coefficients set to 1; both are strict min/max constructions.

The first test run giving reasonable performance in Table 4 is the best MMM case. P–norm interpretation is still superior, however. Hand-crafted queries do slightly better than the best cases where p–values are automatically assigned.

MMM results are a focus of this study and seem fairly easy to interpret. It is clearly unwise to define OR using *max* alone. It is also unwise to define AND using *min* alone. A linear combination of the two gives better results; for OR it is best to emphasize *max* and for AND it is best to emphasize *min*. The best precision occurs when C_{AND} and C_{OR} are both around 0.6 or 0.7. The interaction of these coefficients has been analyzed using regression methods. Table 5 summarizes the values obtained.

Average Precision Results for Different Values of P_{AND} and P_{OR} for P–norm Queries

Table 1

P_{OR}	P_{AND}															
	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0	3.2	3.4	3.6	3.8	4.0
1.0	183	.183	.184	.184	.184	.184	.184	.184	.185	.185	.184	.185	.184	.184	.185	.185
1.2	.184	.184	.184	.184	.184	.184	.185	.185	.184	.185	.185	.184	.184	.184	.184	.184
1.4	.183	.183	.184	.183	.184	.184	.183	.183	.183	.184	.184	.185	.184	.185	.185	.185
1.6	.181	.181	.181	.181	.181	.181	.182	.182	.182	.182	.182	.182	.183	.183	.183	.183
1.8	.179	.179	.180	.179	.179	.179	.180	.180	.180	.180	.180	.181	.181	.182	.182	.182
2.0	.178	.179	.179	.179	.179	.179	.180	.180	.180	.180	.180	.179	.179	.179	.179	.180
2.2	.177	.177	.177	.177	.177	.177	.178	.177	.177	.177	.177	.177	.177	.177	.177	.177
2.4	.176	.176	.176	.175	.175	.175	.175	.175	.175	.175	.175	.175	.175	.175	.176	.176
2.6	.175	.175	.174	.174	.174	.174	.174	.174	.175	.175	.175	.175	.175	.175	.175	.175
2.8	.173	.173	.173	.173	.173	.173	.173	.174	.174	.173	.173	.173	.173	.173	.173	.173
3.0	.173	.172	.173	.172	.172	.173	.173	.173	.173	.173	.172	.172	.172	.172	.172	.172
3.2	.171	.172	.172	.172	.172	.172	.172	.173	.172	.171	.172	.172	.171	.171	.171	.171
3.4	.171	.171	.171	.171	.172	.172	.172	.172	.171	.171	.171	.171	.170	.171	.171	.171
3.6	.170	.170	.170	.170	.171	.171	.171	.171	.171	.171	.170	.170	.170	.170	.170	.170
3.8	.169	.170	.169	.170	.170	.170	.170	.170	.170	.170	.169	.169	.169	.169	.169	.170
4.0	.169	.169	.169	.169	.170	.169	.169	.169	.169	.169	.169	.169	.169	.169	.169	.169

Table 2

Pand	Por	Average Precision
5.0	1.0	.1848
5.0	2.0	.1791
5.0	5.0	.1682
6.0	6.0	.1682
7.0	7.0	.1683
8.0	8.0	.1677
9.0	9.0	.1675
10.0	1.0	.1845
10.0	2.0	.1769
10.0	10.0	.1674
15.0	1.0	.1818
15.0	2.0	.1767
15.0	15.0	.1656
∞	∞	.1180

Table 3. Average precision results for different values of C_{AND} and C_{OR} for MMM queries

C_{OR}	C_{AND}										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.0	.1220	.1291	.1360	.1422	.1505	.1575	.1647	.1659	.1634	.1609	.1180
0.1	.1317	.1358	.1425	.1491	.1557	.1606	.1642	.1667	.1647	.1615	.1180
0.2	.1370	.1435	.1492	.1558	.1602	.1620	.1652	.1684	.1647	.1612	.1170
0.3	.1448	.1490	.1550	.1599	.1641	.1657	.1686	.1685	.1668	.1629	.1166
0.4	.1497	.1536	.1598	.1649	.1663	.1693	.1710	.1702	.1683	.1647	.1162
0.5	.1540	.1572	.1613	.1666	.1673	.1699	.1720	.1724	.1689	.1664	.1177
0.6	.1549	.1587	.1627	.1674	.1668	.1695	.1723	.1718	.1683	.1652	.1188
0.7	.1566	.1595	.1633	.1674	.1678	.1686	.1706	.1715	.1694	.1659	.1188
0.8	.1569	.1594	.1624	.1655	.1672	.1678	.1690	.1703	.1689	.1663	.1192
0.9	.1573	.1598	.1621	.1649	.1652	.1664	.1668	.1677	.1697	.1666	.1188
1.0	.1079	.1082	.1085	.1091	.1090	.1090	.1097	.1103	.1110	.1110	.0369

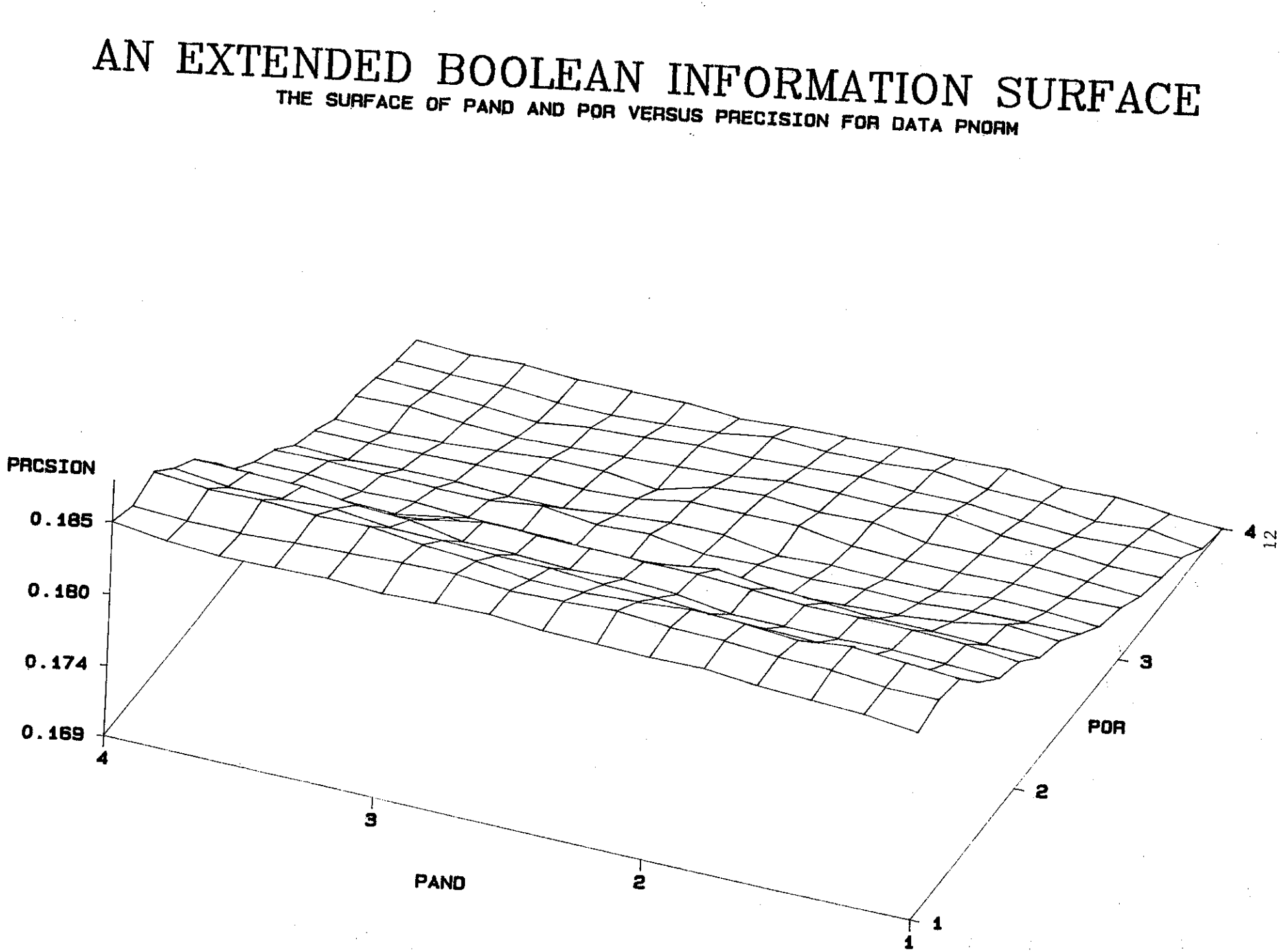

AN EXTENDED BOOLEAN INFORMATION SURFACE
THE SURFACE OF PAND AND POR VERSUS PRECISION FOR DATA PNORM
PRCSION
0.185
0.180
0.174
0.169
4
3
2
1
PAND
1
2
3
4
POR
12

Figure 6

AN EXTENDED BOOLEAN INFORMATION SURFACE

THE SURFACE OF PAND AND POR VERSUS PRECISION FOR DATA PAICE

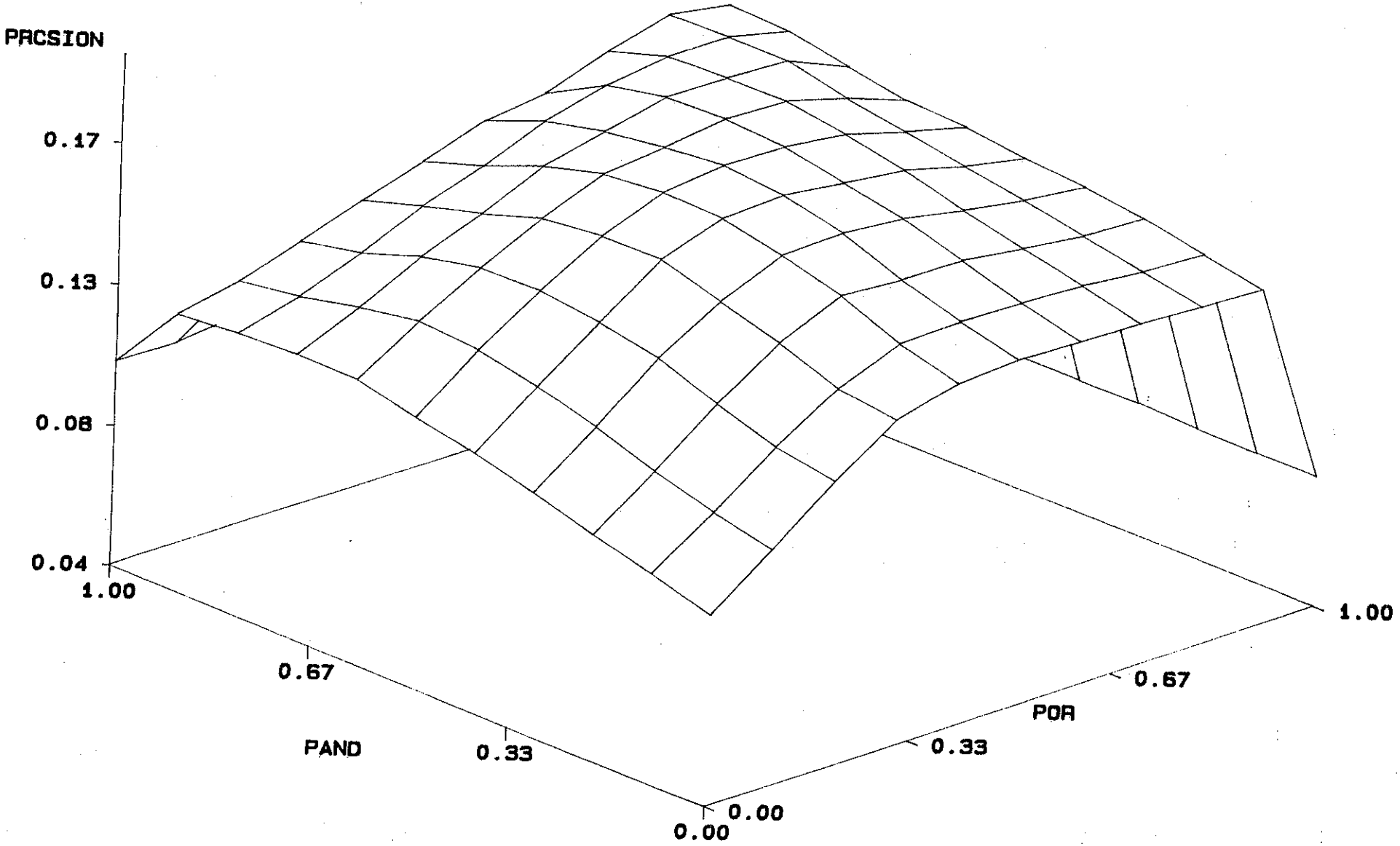


Figure 7

FIGURE 8. P-norm
Av. Precision vs. p_{or} for constant p_{and}

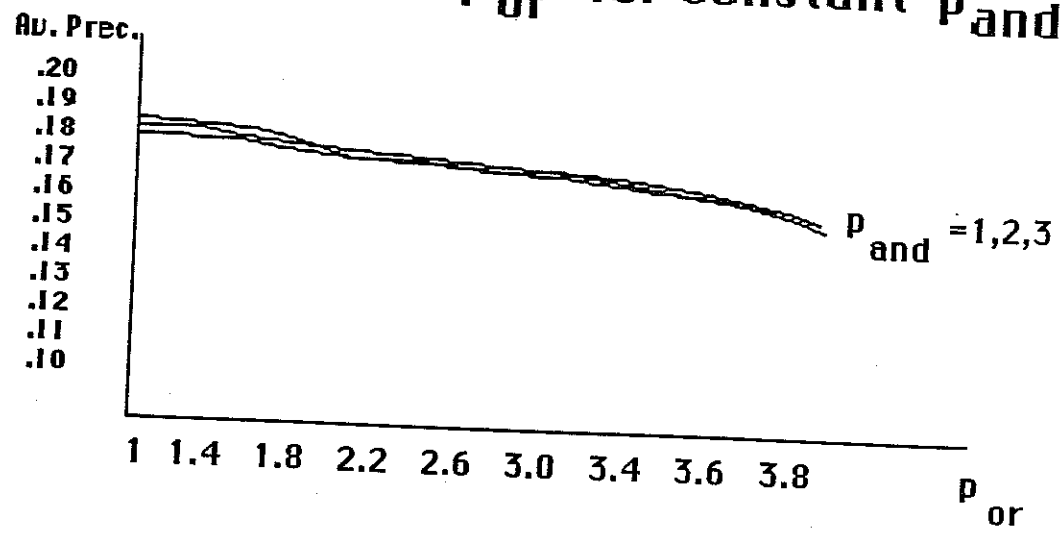


FIGURE 9. P-norm
Av. Precision vs. p_{and} for constant p_{or}

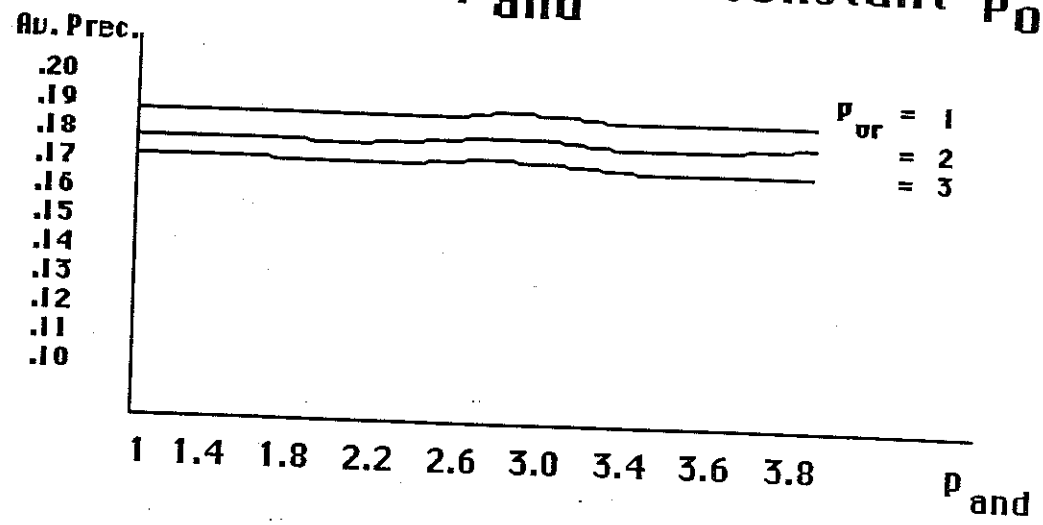


FIGURE 10. MMM
Av. Precision vs. C_{or} for constant C_{and}

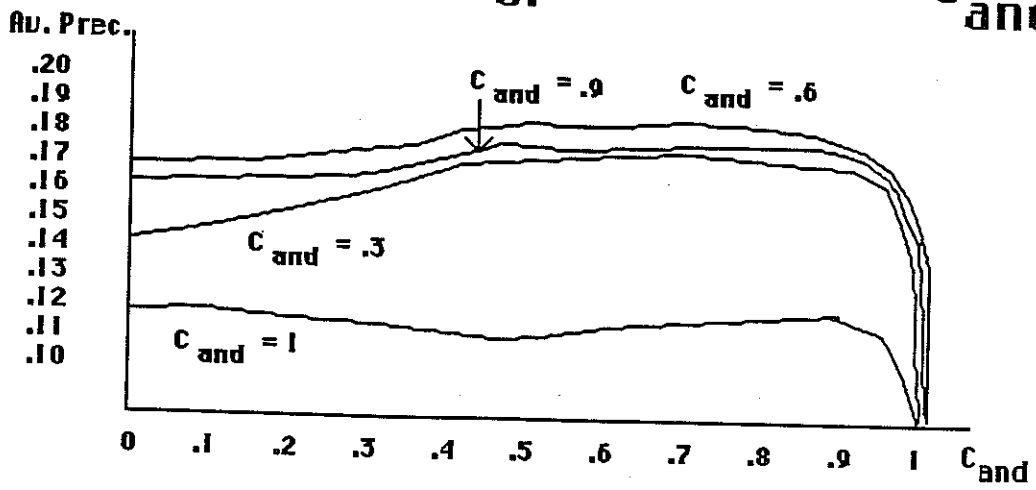


FIGURE 11. MMM
Av. Precision vs. C_{and} for constant C_{or}

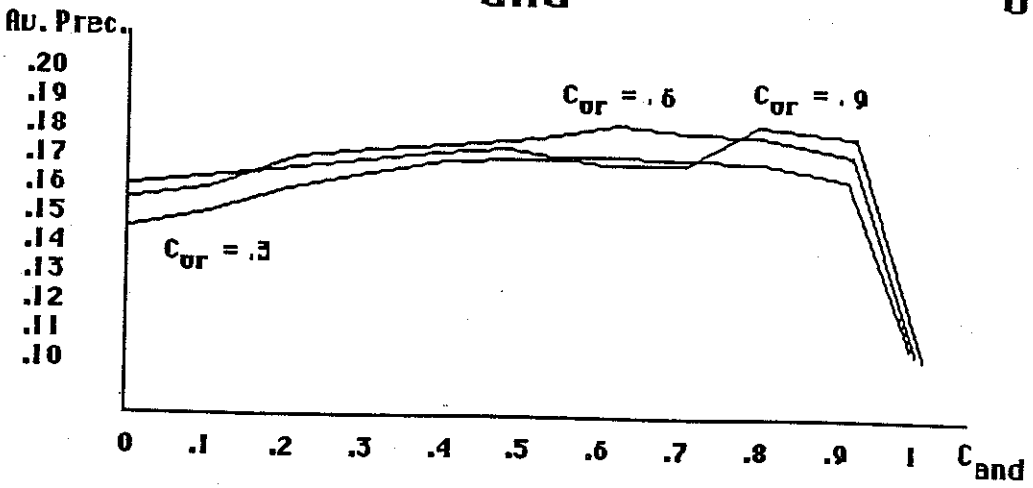


Table 4. Summary of Average Precision Results

AV. PREC.	DESCRIPTION
.104	Boolean retrieval with binary weights, usual operators
.118	P–norm retrieval with p–values set to ∞
.172	MMM with $C_{OR} = 0.6$, $C_{AND} = 0.7$
.185	P–norm for p_{OR}, p_{AND} = (1.2,2.2),(1.4,4.0), etc.
.186	P–norm values assigned heuristically to each query

Table 5. Regression Results for MMM Scheme

R^2	MSE	PRESS	ABS. PRESS	C_p	MODEL
.0254	.000787	.1024	2.71588	503.890	C_{AND}, C_{OR}
.0018	.000806	.1046	2.77894	518.839	C_{AND}, C_{OR}, $C_{AND}C_{OR}$
.6659	.00274	.0365	1.47352	102.485	C_{AND}, C_{OR}, $C_{AND}C_{OR}$, C^2_{AND}, C^2_{OR}
.8194	.000147	.0190	1.1286	3.3169	C^2_{AND}, C^2_{OR}, C^3_{AND}, C^3_{OR}

Clearly there is no good fit until quadratic or cubic terms are included, as one might expect from examining Figure 7. Thus, it is best to interpret both AND and OR fairly strictly, using the MMM scheme.

P–norm results are more complex. It is clear from Figure 6 that there is no dramatic effect on results as p–values change in the small region examined. From Table 2, however, it can be seen that p–values above 5 do cause reduced performance. The heuristics most recently stated in [9] seem to work, as shown by the fact that the hand-tailored queries were (by very small margin) best. Regression analysis (see Table 6 below) shows that a simple, linear model gives a very good fit. Figure 8 makes it clear that the P_{OR} value has the greatest effect, which is also obvious from comments above.

Table 6. Regression Results for P-norm Scheme

R^2	MSE	PRESS	ABS. PRESS	C_p	MODEL
.9640	10.3×10^{-7}	.0026	.2219	409.76	P_{AND}, P_{OR}
.9798	5.805×10^{-7}	.0015	.1554	124.820	P_{AND}, P_{OR}, P_{OR}^2, P_{AND}^2, $P_{AND}P_{OR}$

CONCLUSIONS

Based on a large number of runs using one moderate sized collection of documents and a set of 35 queries with attendent judgments of document relevance, some insights into the use of fuzzy logic for information retrieval have been gained. These further support and expand upon earlier results of Fox, Salton, Paice, Voorhees, Tong, and others.

First, it appears that standard Boolean logic gives lowest average precision. Second, there is a slight improvement when membership functions are used for document terms, and a strict interpretation of operators is still employed. Once the Boolean operators are "softened," however, there is a noticeable improvement.

Of the two techniques for soft Boolean evaluation that are discussed here, in connection with a particular test collection, p–norm did best. Setting parameters for the other scheme, by Paice, is straightforward; one should interpret both AND and OR fairly strictly. Parameters for the p–norm scheme can best be determined by following some heuristics and semantically analyzing the query clauses. However, there is not a great deal of difference among results over the range of small p–values. In general, a very low P_{OR} should be employed and a moderately high P_{AND} should be used.

These findings suggest that:

1) fuzzy set membership functions are very valuable for information retrieval
2) the usual interpretation of AND and OR as *min* and *max*, respectively, is too strict for Boolean queries
3) the p–norm scheme is more effective than the mixed min/max (MMM) scheme advocated by Paice

4) the interaction of parameters used to soften operators follows a number of rules, as can been seen in Tables and Figures herein presented.

Future work will continue these comparisons, using other techniques for interpreting Boolean queries, and validating them further on several other test collections.

ACKNOWLEDGEMENTS

Thanks go to Dr. Matthew Koll of George Mason University for his comments on the regression analysis described. Dr. Philip L. Gatz Jr. of the Virginia Tech Statistics Center prepared the contour plots in Figures 6 and 7, and also carried out a number of supplemental regression runs to help determine the best fits for p–norm and MMM schemes.

REFERENCES

[1] Attar, Rony and Aviezri S. Fraenkel. Local Feedback in Full-Text Retrieval Systems. *J. ACM*, 24(3): pages 397-417, July 1977.

[2] Blair, David C. and M. E. Maron. An Evaluation of Retrieval Effectiveness for a Full-Text Document Retrieval System. *Commun. ACM*, 28(3): pages 289-299, March 1985.

[3] Fox, Edward A. Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types. Dissertation, Cornell Univ., Ithaca, NY, Aug. 1983.

[4] Fox, Edward A. Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographic Concepts. Tech. Report 83-561, Cornell Univ., Dept. of Comp. Sci., Ithaca, NY, Sept. 1983.

[5] Meadow, Charles T. and Pauline A. Cochrane. *Basics of Online Searching*. John Wiley and Sons, New York, 1981.

[6] Paice, C. D. Soft Evaluation of Boolean Search Queries in Information Retrieval. *Inf. Tech.: Res. Dev. Applications*, 3(1): pages 33-42, 1984.

[7] Salton, Gerard, Edward A. Fox and Harry Wu. Extended Boolean Information Retrieval. *Commun. ACM*, 26(11): pages 1022-1036, Nov. 1983.

[8] Salton, Gerard and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[9] Salton, G. and E. Voorhees. Automatic Assignment of Soft Boolean Operators. *Proc. Eighth Annual Intl ACM SIGIR Conf. on R&D in Inf. Ret.*: pages 54-69, June 1985.

[10] Tong, Richard M., Daniel G. Shapiro, Jeffrey S. Dean and Brian P. McCune. A Comparison of Uncertainty Calculi in an Expert System for Information Retrieval. *Proc. IJCAI-83*, August 1983.

[11] Zadeh, L. A. Fuzzy Sets. *Information and Control*, 8: pages 338-353, 1965.