# Computationally Tractable Probabilistic Modeling of Boolean Operators

**Abstract**

The inference network model of information retrieval allows for a probabilistic interpretation of Boolean query operators. Prior work has shown, however, that these operators do not perform as well as the the *pnorm* operators developed in the context of the vector space model. The design of alternative operators in the inference network framework must contend with the issue of computational tractability. We define a flexible class of link matrices that are natural candidates for the implementation of Boolean operators and an $O(n^2)$ algorithm for the computation of probabilities involving link matrices of this class. We present experimental results indicating that Boolean operators implemented in terms of link matrices from this class perform at least as well as *pnorm* operators.

## 1 Introduction

We present a class of operators for the modeling of Boolean queries that is:

- intuitively plausible,
- probabilistically motivated, and
- computationally tractable.

For reasons to be explained, we have called these PIC operators.

In his doctoral dissertation [Tur90], Howard Turtle developed a probabilistic model for information retrieval formulated in terms of a Bayesian inference network. An attractive aspect of the inference network approach is that it provides a direct, natural, computationally efficient, probabilistically motivated method for modeling Boolean query operators. Other approaches to the modeling of Boolean query operators have been tried. As with the inference network, the goal has been to generalize the classical, strict Propositional Logic interpretation of the Boolean operators. The objectives of these generalizations have been twofold:

- to allow for graduated inputs to the Boolean operators so that the representation of documents in terms of vectors of Boolean characteristics can be extended to vectors of feature weights.
- to generate real valued operator output so that dichotomous relevance judgments can be replaced by document ranking in response to user's queries.

A particularly notable success in this pursuit was reported by Salton, Fox and Wu [SFW83, SBF83]. Grounded in the geometric metaphor of the vector space model, they defined a general class of "*pnorm*" operators which extend traditional Boolean operators in a natural way. The experimental results achieved were quite positive.

Recent experiments with INQUERY have shown that the *pnorm* operators yield retrieval performance superior to that of the AND and OR operators as they are currently implemented. Despite these results, the *pnorm* operator computation has not been incorporated in the INQUERY system for want

of a convincing probabilistic justification. This situation has prompted the search for an alternative to the current scheme for modeling operators within the inference network framework that might improve system performance for Boolean queries. A major consideration in this work is the need for the computations associated with the model adopted to respect the computational limitations imposed by modern large scale information retrieval systems. As we shall see, this can be problematic within the confines of the inference network framework.

**Inference Network**

The INQUERY inference network is a Bayesian Network [KP83, Pea88, Cha91] designed for supporting information retrieval. The reader is referred to [TC90] for a description of the general inference network framework.

A Bayesian Network encodes a joint probability distribution. The nodes of the network correspond to random variables. In the INQUERY inference network, each random variable may assume a value of either true or false. The topology of the network is interpreted as encoding a set of conditional independence relations among the variables.

In general, a set of conditional probabilities must be specified for each node of a Bayesian Network. The probabilities for a node are conditioned on the possible combinations of values that may be assumed by the node's *parents*. For each possible combination of values, $x_1, \ldots, x_n, y \in D_1 \times \ldots \times D_n \times D_q$, that may be assumed by the parent nodes, $P_1, \ldots, P_n$, and child node, $Q$, the probability,

$$pr(Q = y \mid P_1 = x_1, \ldots, P_n = x_n)$$

must be specified. These probabilities are conveniently organized in the form of a *link matrix*. Each row of the matrix corresponds to a possible value for the child variable, and each column to a possible combination of values for the parent nodes. Each column, then, gives the probability distribution over $D_q$, the possible values of $Q$, conditioned on one possible setting, $x_1, \ldots, x_n$, for the set of parent variables. In the case of the inference network, a $2 \times 2^n$ matrix is needed since the range of each variable is $\{\text{true}, \text{false}\}$.

**Boolean queries**

In [Tur90], Turtle describes how the inference network can be used to model Boolean queries. Link matrices can be defined that are natural interpretations for the Boolean operators and can be evaluated very efficiently. They may be questioned on two counts, however.

First, the interpretation given to the AND and OR operators, while natural, might be considered overly strict. Faced with a user need expressed in terms of AND, it is likely that most people's belief system would be inclined toward assigning greater, albeit perhaps only slightly greater, probability to the query being satisfied for those events for which a greater number of parents is true. The INQUERY AND operator does not do this. The belief assigned to a document for which 9 of 10 concepts are known to be present with 100% certainty, while 1 of the concepts is known with certainty to be absent, will be the same as that assigned to a document for which all 10 concepts are known with certainty to be absent[1]. While this is, perhaps, a reasonable interpretation of what a user should mean when she employs the AND operator, it probably does not correspond to what she is really trying to express. An analogous argument can be made with regard to the OR operator.

The second cause for concern with regard to the operators, as they are currently defined, is that experiments have shown that they are not as effective as might be hoped for. Recently, Leah Larkey carried out a series of experiments to test the efficacy of the *pnorm* operators in the context of the inference network [Lar96]. She found that system performance on four sets of Boolean queries, as

[1] While this extreme case cannot occur in the current implementation of INQUERY, all belief calculations involving the AND operator are affected by the fact that the operator is modeled in this way

measured by average precision, was improved by replacing the inference network Boolean operator calculation used in INQUERY by the *pnorm* AND and OR operator computations.

**Pnorm**

Working within the context of the vector space model, Salton, et al in [SBF83] and [SFW83]. were able to extend conventional Boolean retrieval in a way that allowed for: 1) the weighting of document features, and 2) the production of ranked output. They normalized weights so that all vector components were in the range of 0.0 to 1.0 and, hence, all vectors lie in the unit hypercube. Intuitions based on a geometric view of information retrieval led them to consider the output of an OR operator as a measure of *distance* from the point $< 0.0, 0.0, \ldots, 0.0 >$. So, for the query

$$\texttt{\#or}(t_1, \ldots, t_n)$$

a document with term weights $w_1, \ldots, w_n$ for terms $t_1, \ldots, t_n$, would receive a score of,

$$(\frac{w_1^2 + \ldots + w_n^2}{n})^{\frac{1}{2}} \tag{1}$$

This distance measure can be viewed as a special case of the more general vector norm measure [Ort72] [2]:

$$(\frac{w_1^p + \ldots + w_n^p}{n})^{\frac{1}{p}}$$

This operator can be considered or-like. When all weights are 0.0, it will produce a value of 0.0. When all weights are 1.0, it will produce a value of 1.0. When a variety of weights are presented as inputs, the output is more heavily influenced by the weights at the higher end of the scale. The result is that lower values can be *overpowered*, to a degree, by a single large value. This is reasonable behavior for a generalization of an OR operator. We may presume that, on specifying an OR, the user is indicating that lack of evidence with respect to most of the terms can be downplayed in the presence of strong evidence with regard to just one of the terms.

For smaller values of $p$, the operator is less or-like. At $p = 1$, the operator degenerates to a simple average, and the Boolean structure of the query is essentially ignored. As $p$ grows, more and more weight is given to the larger input components. In the limit, the calculation is equivalent to that of the MAX operator, which is the standard operator used in models of information retrieval based on fuzzy-set theory [NKM77, WK79, Boo80, Boo81, BK81], and corresponds to a strict Boolean OR when the input components are limited to strict Boolean values.

For concreteness, we have concentrated on the OR operator. The AND operator, given by:

$$1 - (\frac{(1-w_1)^p + \ldots + (1-w_n)^p}{n})^{\frac{1}{p}}$$

can be understood as the dual of the OR operator, and all aspects of the above discussion have their counterpart with respect to AND as well.

[SFW83] report experimental results in which use of the *pnorm* model consistently improves performance over basic Boolean retrieval on four relatively small collections. More recently, Joon Ho Lee has run experiments on the larger TREC (Text REtreival Conference) sub-collection, the Wall Street Journal Disk 2 [Lee95]. He shows that *pnorm* performs favorably compared to a variety of other formalisms. Nonetheless, it is difficult to see how the *pnorm* operators can be given a probabilistic interpretation. For those interested in exploring probabilistic IR models, the need to explore alternatives remains.

[2] It should be said that the formula presented in [SFW83] is more general than the one shown here in that it allows for the weighting of query terms as well as document terms. Nonetheless, the experiments reported there utilize only binary query weights. As this aspect of the formula is not relevant to the work discussed here, we focus our attention on a somewhat simplified version of the formula.

## 2 PIC Matrices and the PIC-EVAL Algorithm

A wide range of functions can be represented as link matrices. Unfortunately, the evaluation of an arbitrary link matrix for an arbitrary set of input probabilities requires $O(2^n)$ floating point operations. In [Tur90], Turtle shows that closed form expressions exist for some matrices, such that they may be evaluated in time linear in the number of parents, for arbitrary inputs. The matrices used for Boolean operators have that characteristic. It is not difficult to show that the probability that the child node is true can be given by:

$$pr(Q \text{ is true}) \quad = \quad p_1 p_2 \cdots p_n$$

for the AND operator, and

$$pr(Q \text{ is true}) \quad = \quad 1 - (1-p_1)(1-p_2)\ldots(1-p_n)$$

for OR, where $p_i$ is the probability that parent $P_i$ is true.

### The PIC matrices

The desire to explore new possibilities for Boolean query operators motivates the search for wider classes of computationally tractable link matrices. A natural candidate for consideration is the class of matrices for which the conditional probability that the child node is true is determined solely by the number of parents that are true, independent of which of them happen to be true and which are false.

We will say that a link matrix satisfies the *parent indifference criterion*, or simply that it is a PIC matrix if, given the parent nodes, $P_1, \ldots, P_n$:

$$\forall R_1, R_2 \subseteq \{P_1, \ldots, P_n\}: \quad |R_1| = |R_2| \Rightarrow \alpha_{R_1} = \alpha_{R_2}$$

where for each $R \subseteq \{P_1, \ldots, P_n\}$:
$\alpha_R$ is the matrix coefficient associated with the event that
the $P_i$ belonging to $R$ are true and the $P_i$ not in $R$ are false

A PIC matrix for $n$ parents requires the specification of $n+1$ parameters, $\alpha_0, \ldots, \alpha_n$, as compared to the $2^n$ parameters required for an arbitrary link matrix. Each $\alpha_i$ specifies the common value given by $\alpha_R$ for each $R \subseteq \{P_1, \ldots, P_n\}$ such that $|R| = i$. The sequence of link matrix coefficients, $\alpha_0, \ldots, \alpha_n$, can be viewed as a function from the integers $\{0, 1, \ldots, n\}$ to the interval $[0, 1]$.

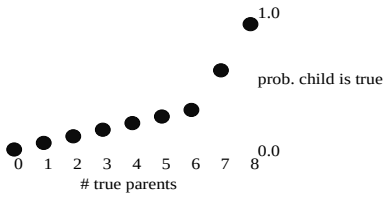


Figure 1: PIC matrix as non-decreasing function

It is appropriate to note that we have in mind, and will be exploring, coefficients corresponding to non-decreasing functions, such as that shown in figure 1. Viewing the *pic* matrices as operators for the combination of evidence, our interest is in those operators for which more pieces of individual evidence (i.e. greater number of true parents) translates to greater probability that the combined evidence is present. Nonetheless, the PIC matrix class is not limited to such functions.

The matrix used for the INQUERY SUM operator satisfies the parent indifference criterion. The coefficients for the general $n$-ary SUM matrix are $0, \frac{1}{n}, \frac{2}{n}, \ldots \frac{n-1}{n}, 1$ as shown graphically in figure 2a.

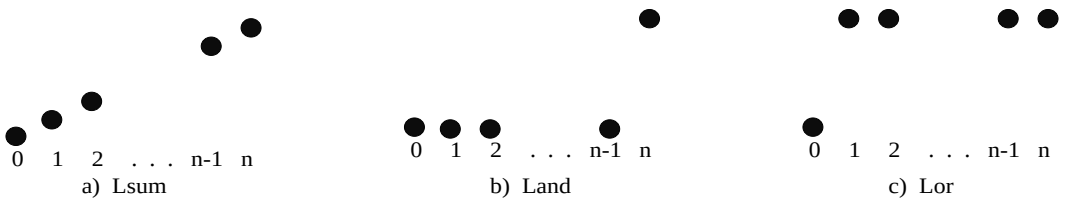


Figure 2: view of SUM, AND, OR as functions of number of true parents

If the probabilities are viewed as weights of evidence, the matrix can be viewed as an operator that averages these weights. That is:

$$pr(Q \text{ is true}) = \frac{p_1 + p_2 + ... + p_n}{n}$$

The matrices used for the AND and OR Boolean operators also meet the parent indifference criterion, as shown in figures 2b and 2c.

In [Tur90], a fourth type of matrix, the *weighted-sum* matrix is shown to be computable in linear time as well. This matrix does not satisfy the parent indifference criterion, but is shown in [Gre96] to satisfy a generalization of this criterion which contemplates a relative weighting of the importance of the parent nodes in determining the probability that the child node is true. A simple extension of the PIC-EVAL algorithm allows for the calculation of probabilities involving these *weighted* PIC matrices in $O(n^2)$ time as well.

inputs:
- $n+1$ coefficients, $\alpha_0, \ldots, \alpha_n$, of a *pic* matrix
- probabilities $p_1, \ldots, p_n$, for the $n$ parent nodes.

output:
- the probability that the child node is true.

PIC-EVAL:

for $j = 0, \ldots, n$
  $\alpha[0, j] = \alpha_j$
for $i = 1, \ldots, n$
  for $j = 0, \ldots, n - i$
    $\alpha[i, j] \leftarrow \alpha[i-1, j] * (1 - p_i) + \alpha[i-1, j+1] * p_i$
return $\alpha[n, 0]$

Figure 3: PIC-EVAL algorithm

**The PIC-EVAL algorithm**

The PIC matrices would not constitute a useful class if they could not be evaluated in better than exponential time. Figure 3 shows an algorithm for the efficient evaluation of matrices that satisfy the parent indifference criterion. Starting with the original link matrix, $L_0$, PIC-EVAL, in effect, generates a sequence of smaller and smaller link matrices, $L_1, L_2, \ldots, L_n$. In the process, it eliminates from consideration each probability in turn (figure 4).

To begin, the probability associated with parent node, $P_1$, is fixed and the matrix, $L_0$, with $n+1$ coefficients and $n$ parents (figure 4a) is converted to an $n$ coefficient link matrix with $n-1$ parents (figure 4b). As shown in figure 4, each matrix, $L_i$, connects with one fewer parent node than the previous one. Corresponding to this, the coefficients, $\alpha_{i,0}, \alpha_{i,1}, \ldots, \alpha_{i,n-1}$, of each matrix are fewer

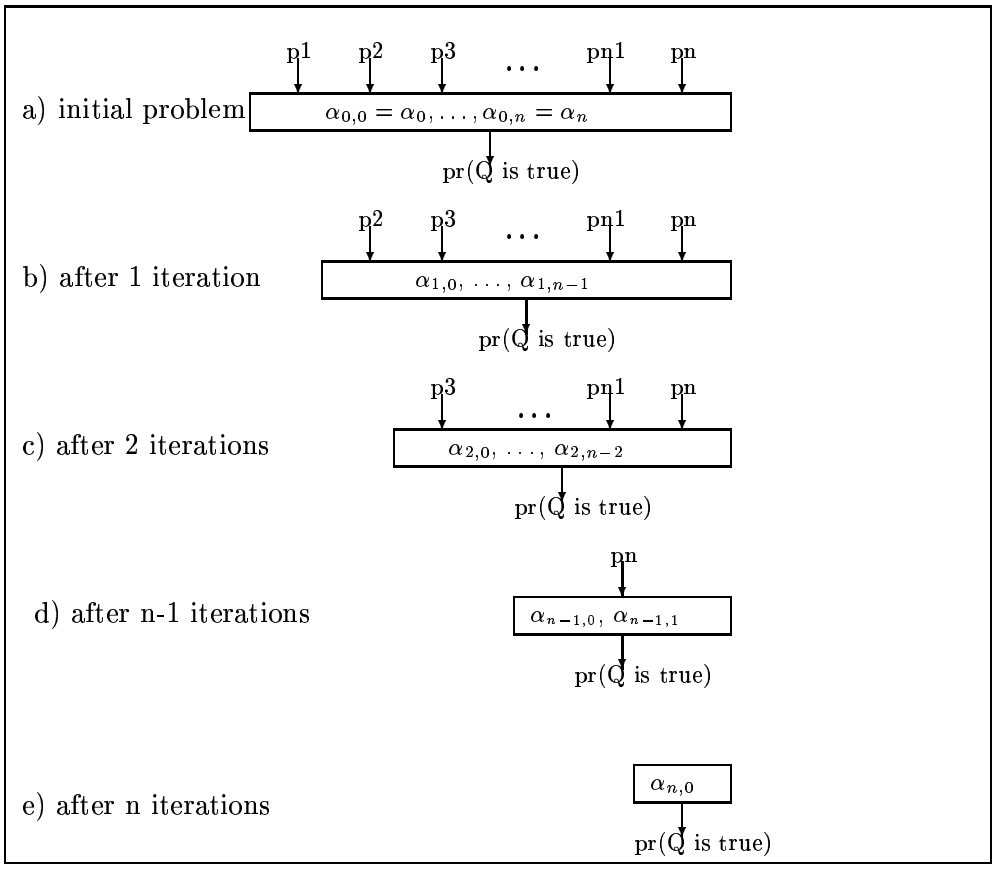


Figure 4: iterations in the evaluation of a *pic* matrix

as well. The matrices, $L_i$, that result from this sequence of transformations are equivalent to the original matrix, $L_0$, in the following sense:

> *for any set of probabilities for parents, $P_{i+1}$, ..., $P_n$, $L_i$ yields the same pr(Q is true) that $L_0$ would produce given the same probabilities for $P_{i+1}$, ..., $P_n$, together with the probabilities $p_1, \ldots p_i$ for $P_1, \ldots, P_i$.*

After $n$ iterations we arrive at $L_n$ with exactly one coefficient, $\alpha_{n,0}$, as seen in figure 4e. Now that all parent probabilities have been accounted for, this coefficient can be interpreted as the desired probability that the child is true. A formal proof of the correctness of the PIC-EVAL algorithm can be found in [Gre96].

The PIC-EVAL algorithm is executed in $O(n^2)$ time. The initialization requires $O(n)$ and the main loop is executed precisely $\sum_{i=0}^{n} \sum_{j=0}^{n-i} 1 = O(n^2)$ times. Also, the constant factor is small. On top of the base iteration control overhead, two multiplications and one addition are required in each iteration. Although, for the purposes of exposition, the algorithm has been shown as requiring $O(n^2)$ space as well as time, $O(n)$ space is easily achieved since only one row's worth of cells need be maintained at any one time.

The PIC-EVAL algorithm can also be optimized for a well-defined, and potentially useful, subclass of the PIC matrices. These are the matrices that are piecewise linear when viewed as functions of the number of true parents. For families of matrices where all but one of the linear pieces of the function is of constant width, evaluation requires only O(n) time. The reader is referred to [Gre96] for details.

# 3 Experimentation

In order to test the potential of the PIC matrices as candidates for the implementation of Boolean query operators, a number of experiments were run with the INQUERY system. For these experiments, two test collections were used: the INSPEC collection and volume 1 of the TIPSTER collection. For the TIPSTER collection, tests were run using two Boolean versions of queries 51 to 100. In this paper, we shall refer to these two sets of queries as *qt1* and *qt2*. Two Boolean query sets, which we will call *qi1* and *qi2*, were also used for testing with the INSPEC collection. The four query sets had been created previously for other IR experiments run with the TIPSTER [BCCC93] and INSPEC [RC19] data. For all experiments discussed in this paper, the INQUERY system using the AND and OR operators as defined in [Tur90] shall be considered the baseline system for purposes of comparison. For these query sets, experiments have shown that *pnorm* operators perform from 7% to 28% better than the baseline system in terms of average precision. Our goal was to achieve similar performance using PIC matrices.

A number of different types of PIC matrices were tried using the standard INQUERY formula for estimating the probability that a document is *about* a concept:

$$
\begin{aligned}
& 0.4 + 0.6 \times \frac{tf}{0.5 + 1.5 \times \frac{dl}{\bar{dl}}} \times \frac{log(\frac{dc+0.5}{df})}{log(dc+1.0)} \\
& \text{where} \quad dl = \text{doc. length (no. of tokens in doc.)} \\
& \qquad\quad \bar{dl} = \text{avg. doc. length (over collection)} \\
& \qquad\quad dc = \text{doc. count} = \text{no. of docs in collection} \\
& \qquad\quad df = \text{doc. freq.} = \text{no. of docs containing term}
\end{aligned}
\tag{2}
$$

Although we were able to realize improvements over the baseline system, it was not possible to obtain results consistently on a par with those produced using the *pnorm* operators. Analysis of the link matrix computations led us to believe that the difficulty might lie in the value used for the *default belief*. In (eq. 2), 0.4 is this default belief: the probability of a document being about

the concept of interest that will be assigned when the associated term count is zero. Experiments with the default belief left at 0.4 were somewhat disappointing. With the default belief set to 0.0, however, it seemed that robust behavior had been obtained, at a level of performance comparable to that of the *pnorm* operators.

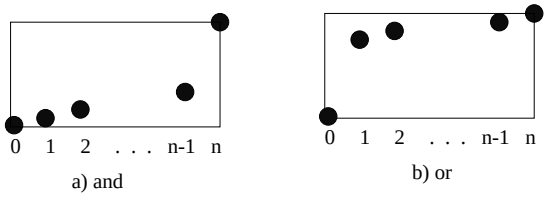


Figure 5: *sloped* PIC matrices for Boolean operators

The experiments described here concentrated on the use of PIC matrices that varied in a simple way from the AND and OR operators used in the current INQUERY system. The standard AND operator can be viewed as a linear sequence of $n-1$ points, $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$, together with the coefficient $\alpha_n = 1.0$. We can think of the sequence of $n-1$ coefficients as rising with a slope of 0.0. In these experiments we consider PIC matrices for the AND operator that are of the same form, but for which the slope may rise with a slope of $\gamma_{and} > 0$. Presumably, the rise will be gradual, with $\gamma_{and}$ set to a fairly small fraction, as shown in figure 5a. To simplify the experimentation, the slope was maintained constant over the entire family of AND operators. That is, for a given experiment, the coefficients for each different arity of the AND operator were determined based on the same setting of $\gamma_{and}$.

The PIC matrix candidates for the OR operators were defined in an analogous manner as is shown in figure 5b. As with the AND operator, $\alpha_0$, and $\alpha_n$ are fixed at 0.0 and 1.0, respectively. In the case of the OR operator, $\gamma_{or}$ corresponds to the rate of decrease in the values of the coefficients as the number of true parents decreases in the sequence, $\alpha_n, \alpha_{n-1}, \ldots, \alpha_2$.

| | $\gamma_{or} = 0.0$ | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|---|
| $\gamma_{and} = 0.2$ | +12.2 | +14.3 | +15.3 | +16.6 | +16.0 | +14.5 |
| 0.4 | +14.8 | +16.9 | +17.8 | +18.7 | +17.5 | +14.4 |
| 0.6 | +16.0 | +17.6 | +18.2 | +19.0 | +17.6 | +14.9 |
| 0.8 | +16.7 | +18.7 | +18.9 | +19.4 | +17.6 | +15.4 |
| 1.0 | +18.9 | +20.0 | +20.3 | ⋆ +20.4 | +18.4 | +15.5 |

(* greatest improvement)

Table 1: *qi1* queries on INSPEC: PIC with $\gamma_{and}, \gamma_{or} \leq 1.0$

Experiments were run first with the INSPEC collection. Table 1 shows the percent improvement in average precision with respect to the baseline system, for different settings of the $\gamma_{and}$ and $\gamma_{or}$ parameters. For all values of $\gamma_{and}$ tested (including values not shown in the above table), optimal performance was achieved with $\gamma_{or}$ set to 0.8. For all values of $\gamma_{and}$, performance improves monotonically with increasing values of $\gamma_{and}$, not reaching a maximum value until $\gamma_{and}$ is at 1.0. This is surprising because AND operators with greater values of $\gamma_{and}$ can be interpreted as operators that increasingly ignore structure. At $\gamma_{and} = 1.0$, structure is ignored entirely and the AND operator degenerates to the SUM operator, which simply averages the input probabilities.

The consistent peak of performance at $\gamma_{and} = 1.0$ immediately suggests testing the performance for values of $\gamma_{and} > 1.0$. Table 2 shows that for all values of $\gamma_{or}$, performance continues to improve as $\gamma_{and}$ increase until $\gamma_{and}$ reaches 3.0. This is highly counterintuitive. At values of $\gamma_{and}$ above 1.0, the PIC matrix has not only been divested of characteristics normally associated with an AND operator, but becomes more and more or-like in nature.

| | $\gamma_{or}$ = 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|---|
| $\gamma_{and}$ = 1.0 | +18.9 | +20.0 | +20.3 | +20.4 | +18.4 | +15.5 |
| 2.0 | +21.6 | +23.9 | ⋆ +26.1 | +24.9 | +22.3 | +17.6 |
| 3.0 | +23.6 | +25.6 | ⋆ +26.1 | +25.3 | +22.4 | +17.1 |
| 4.0 | +23.1 | +25.1 | +25.9 | +24.4 | +21.1 | +15.3 |
| 5.0 | +23.2 | +25.1 | +25.9 | +24.5 | +21.2 | +15.4 |
| 6.0 | +23.1 | +25.1 | +25.9 | +24.5 | +21.1 | +15.4 |
| 7.0 | +23.1 | +25.1 | +25.9 | +24.5 | +21.1 | +15.4 |

Table 2: *qi1* queries on INSPEC: PIC with $\gamma_{and} \geq 1.0$

| | $p_{or}$ = 1.0 | 2.0 | 3.0 | 4.0 | 5.0 | 6.0 | 7.0 |
|---|---|---|---|---|---|---|---|
| $p_{and}$ = 1.0 | +15.5 | +16.1 | +16.0 | +15.7 | +15.6 | +16.0 | +16.4 |
| 2.0 | +16.4 | +17.0 | +17.0 | +17.1 | +17.7 | +17.1 | +16.1 |
| 3.0 | +16.6 | +17.0 | +17.2 | +17.8 | +17.9 | +17.1 | +16.2 |
| 4.0 | +16.7 | ⋆ +18.2 | +18.0 | +17.7 | +17.5 | +16.8 | +16.7 |
| 5.0 | +17.6 | +17.8 | +17.2 | +16.6 | +16.4 | +15.8 | +15.5 |
| 6.0 | +16.4 | +16.5 | +15.7 | +15.5 | +15.5 | +15.0 | +14.3 |

Table 3: *qi1* queries on INSPEC: *pnorm*

Table 3 shows the performance of the *pnorm* operator for various values of $p_{and}$ and $p_{or}$, the settings of the parameter, $p$, for AND and OR respectively. We see that, as with the PIC matrix operators, the *pnorm* operators perform robustly over a wide range of parameter settings. Performance of the PIC matrix operator is clearly superior to that of the *pnorm* operators for this set of queries against the INSPEC collection. Peak performance improvement for the PIC matrix version of the operators reaches 26.1% as compared to 18.2% for the *pnorm* version. Tables 4 and 5 show the comparative performance for, *qi2*, an alternative Boolean formulation of the same queries, against the same collection. Performance is similar to that of the previous query set with the optimal performance of the PIC operators slightly lower at 22.8% improvement as compared to 18.0% for *pnorm*.

| | $\gamma_{or}$ = 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|---|
| $\gamma_{and}$ = 1.0 | +18.6 | +19.5 | +20.0 | +20.1 | +18.9 | +14.3 |
| 2.0 | +21.4 | +22.0 | +22.4 | +22.8 | +20.7 | +14.1 |
| 3.0 | +21.5 | +22.3 | +22.6 | +22.7 | +20.1 | +13.3 |
| 4.0 | +21.7 | +22.5 | ⋆ +22.8 | ⋆ +22.8 | +20.1 | +12.5 |
| 5.0 | +21.6 | +22.5 | ⋆ +22.8 | ⋆ +22.8 | +20.1 | +12.5 |
| 6.0 | +21.6 | +22.5 | ⋆ +22.8 | ⋆ +22.8 | +20.1 | +12.5 |
| 7.0 | +21.6 | +22.5 | ⋆ +22.8 | ⋆ +22.8 | +20.1 | +12.5 |

Table 4: *qi2* queries on INSPEC: PIC

A similar set of experiments were run on the larger TIPSTER collection. Table 6 summarizes the performance of the two approaches. We see here that both classes of operators are capable of significantly outperforming the baseline system on both query sets. On the first query set, the best parameter setting for the *pnorm* operators slightly outperforms the best settings for the PIC operators. On the second query set, it is the best settings of the PIC operators that yield superior performance. The smaller improvements for both systems on the second query set is due to the better performance of the baseline system. It is unclear why the baseline system is apparently so much more sensitive to the differences in the two query formulations than the *pnorm* and PIC versions of the system.

Surprisingly, in light of the previous results, this best performance for the PIC system on the second query set is obtained with the $\gamma_{and}$ coefficient set as low as 0.1. To date, we have been unable to

| | $p_{or}$ = 1.0 | 2.0 | 3.0 | 4.0 | 5.0 | 6.0 | 7.0 |
|---|---|---|---|---|---|---|---|
| $p_{and}$ = 1.0 | +14.3 | +15.7 | +16.3 | +16.8 | +17.1 | +17.5 | +16.9 |
| 2.0 | +15.2 | +16.4 | +17.5 | +17.6 | +17.9 | +17.8 | +16.8 |
| 3.0 | +15.5 | +16.3 | +16.9 | +17.2 | +17.9 | +17.7 | +16.9 |
| 4.0 | +15.9 | +16.9 | +17.2 | +17.5 | ⋆ +18.0 | ⋆ +18.0 | +17.1 |
| 5.0 | +15.2 | +16.4 | +16.7 | +16.9 | +17.4 | +17.5 | +16.7 |
| 6.0 | +14.9 | +15.4 | +15.8 | +16.5 | +16.7 | +16.8 | +16.1 |

Table 5: *qi2* queries on INSPEC: *pnorm*

explain this phenomenon. We are, however, quite interested in understanding what is happening here. Insight into this behavior should, we think, lead to important insights into the interpretation of Boolean query operators in general.

Table 6 also compares the two operators with *pnorm* settings of $p_{and}$ = 6.0, $p_{or}$ = 3.0 and PIC settings of $\gamma_{and}$ = 2.0, $\gamma_{or}$ = 0.6. These parameter settings were chosen so as to give a best overall performance profile for each of the operator classes. In general, the optimal settings for the *pnorm* operator in these experiments tend to be somewhat higher than those reported in [SFW83]. For comparable experiments they found that values between 1.0 and 2.0 produced best results. However, in their experiments they did not attempt to vary the value of $p$ for AND and that for OR independently. It is possible that had they allowed for independent settings, a higher value for one, or perhaps even both, of the parameters might have produced better results.

| | base line | *pnorm* $p_{and}$ /$p_{or}$ | avg. prec. | % imp. | PIC $\gamma_{and}$ /$\gamma_{or}$ | avg. prec. | % imp. | *pnorm* | % imp. | PIC | % imp. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *qi1* | 26.5 | 4.0/2.0 | 31.3 | 18.2 | 2.0/0.4 | 33.4 | 26.1 | 30.7 | 15.7 | 33.1 | 24.9 |
| *qi2* | 25.6 | 4.0/5.0 | 30.2 | 18.0 | 4.0/0.4 | 31.4 | 22.8 | 29.6 | 15.8 | 31.4 | 22.8 |
| *qt1* | 20.8 | 9.0/1.0 | 26.8 | 28.8 | 2.0/0.8 | 26.6 | 20.8 | 26.5 | 27.3 | 26.4 | 26.9 |
| *qt2* | 25.0 | 10.0/1.0 | 26.9 | 7.8 | 0.1/0.6 | 27.5 | 9.8 | 25.0 | 6.8 | 26.4 | 5.5 |

Table 6: comparison of PIC and *pnorm* for all four query sets

# 4 Conclusions

From the work reported here, we may conclude that combining functions with a well-defined probabilistic interpretation can be associated with Boolean query operators, which

- appear to perform at least as well as the *pnorm* operators of Salton, et al, and
- can be realized at reasonable computational cost.

**Probabilistic interpretation**

The definition of the *pnorm* functions is an outgrowth of intuitions grounded in the vector space model of information retrieval [Sal89]. The PIC matrices, on the other hand, are the result of viewing the scoring of documents from a probabilistic standpoint. Hence, comparative behavior on varying collections and/or varying query sets can be studied from a probabilistic vantage point. Attempts to improve overall performance can be guided by researchers' intuitions as to the probabilities involved. Coherent techniques for combining Boolean operators with others, such as proximity or phrase operators, can be developed in accord with the laws of probability theory.

**Retrieval Performance**

From the four different query set formulations studied, it appears that combining functions based on PIC matrices perform as well as the *pnorm* functions. For the researcher inclined toward the

probabilistic approach, this is comforting. The definition of the *pnorm* operators is an excellent example of how a mathematical model, in this case the vector space model, can guide the researcher toward the development of fruitful ideas. We have shown here that, at least as far as the current state of the art with respect to Boolean operators is concerned, a probabilistic theory of information retrieval can be equally beneficial in this regard. It is our belief, and apparently that of others in the "probabilistic camp" (for example, see [Coo94] for an interesting exposition of the issues), that in the long run, models founded in probability theory will prove to be more fruitful in this regard.

The results obtained here also suggest that some previous experimental results might merit a second look. For example, [Tur94] reports that natural language queries consistently outperform Boolean queries for experiments run on legal collections. The difference observed, however, is within the range of the improvements we have been able to obtain, suggesting that more effective implementations of the Boolean query operators might close the apparent gap.

**Computational Efficiency**

Probability calculation involving PIC matrices can be realized in $O(n^2)$ time. The CPU time for the PIC matrix version of INQUERY was compared against the baseline system for each of the query sets discussed in this paper. An increase of from 35% to 65% in CPU time was observed. This would seem to be a reasonable price to pay for the increases in performance that can be realized, especially since the percentage increase in overall response time can be expected to be significantly smaller when I/O time is factored in. Exactly what that overall increase would be will depend, of course, on the characteristics of the particular hardware used to run the system [3].

**Future work**

The experimentation reported here indicates that PIC matrices can be advantageously applied to the modeling of Boolean queries. Three ways in which we are considering extending this work are:

*alternative sub-classes of the* PIC *matrices:* The PIC matrices cover a wide range of functions which can be chosen for modeling belief-combination operators. The definition of an operator family requires the specification of $n + 1$ parameters for each arity, $n = 2, 3, \ldots$. We have had success with a subclass of the PIC matrices that can be specified with two parameters. Other sub-classes, resulting from other parameterizations may prove to be superior.

*experimenting with default probabilities:* We have obtained our best results by eliminating the default setting of 0.4 in favor of a 0.0 setting. It is possible that default settings at some value somewhat above 0.0 will yield small, but consistent, performance gains.

*correlating beliefs with long term frequencies:* If the design of the Boolean operators follows probabilistic intuitions, it is reasonable to expect that their success in practice will be correlated with the extent to which the values being manipulated correspond to probabilities in the real world. It should be instructive to analyze the behavior of the operators after performing some kind of regression to convert the document scores into scores more accurately reflecting observed frequencies of relevance for large text collections.

# 5 Acknowledgments

The authors would like to express their gratitude to Leah Larkey whose original experimentation with *pnorm* operators in the context of the INQUERY inference network was the starting point for the experimentation reported here, This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement

[3] As mentioned in section 2, a linear time version of the PIC-EVAL algorithm is applicable when the PIC matrix is a piecewise linear function. Although this more efficient version of the algorithm would be applicable to the PIC matrices utilized for these particular experiments, the general PIC matrix algorithm was used. The times reported correspond to the general version of the algorithm.

number EEC-9209623. This material is also based on work supported in part by United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsors.

# References

[BCCC93] N. J. Belkin, C. Cool, W. B. Croft, and J. P. Callan. The effect of multiple query representations on information retrieval system performance. In Robert Ksorfhage, Edie Rasmussen, and Peter Willett, editors, *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 339–346, June 1993.

[BK81] D. A. Buell and D. H. Kraft. Threshold values and Boolean retrieval system. *Information Processing & Management*, 17(3):127–136, 1981.

[Boo80] A. Bookstein. Fuzzy requests: an approach to weighted Boolean searches. *Journal of the American Society for Information Retrieval*, 31(4):240–247, July 1980.

[Boo81] A. Bookstein. A comparison of two systems of weighted Boolean queries. *Journal of the American Society for Information Retrieval*, 32(4):275–279, July 1981.

[Cha91] E. Charniak. Bayesian networks without tears. *AI Magazine*, 12(4):50–63, April 1991.

[Coo94] William S. Cooper. The formalism of probability theory in IR: A foundation or an encumbrance. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 242–248, Dublin, Ireland, July 1994.

[Gre96] Warren R. Greiff. Computationally tractable, conceptually plausible classes of link matrices for the inquery inference network. Technical Report CmpSci TR-96-66, University of Massachusetts, Amherst, Massachusetts, September 1996.

[KP83] J. H. Kim and J. Pearl. A computational model for combined causal and diagnostic reasoning in inference systems. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 190–193, Karlsruhe, West Germany, August 1983.

[Lar96] Leah S. Larkey. personal communication, 1996.

[Lee95] J. H. Lee. Analyzing the effectiveness of extended Boolean models in Information Retrieval. Technical Report TR95-1501, Cornell University, 1995.

[NKM77] T. Noreault, M. Koll, and M. J. McGill. Automatic ranked output from Boolean searches in SIRE. *Journal of the American Society for Information Retrieval*, 28(6):333–339, 1977.

[Ort72] J. M. Ortega. *Numerical Analysis: A Second Course.* Academic Press, New York, 1972.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[RC19] T. B. Rajashekar and W. Bruce Croft. Combining automatic and manual index representations in probabilistic retrieval. *Journal of the American Society for Information Retrieval*, 46(4):272–283, 19.

[Sal89] Gerard Salton. *Automatic text processing : the transformation, analysis, and retrieval of information by computer.* Addison-Wesley Publishing Company, Reading, MA, 1989.

[SBF83] G. Salton, C. Buckley, and E. A. Fox. Automatic query formulations in information retrieval. *Journal of the American Society for Information Retrieval*, 34(4):262–280, July 1983.

[SFW83] Gerard Salton, Edward A. Fox, and Harry Wu. Extended Boolean information retrieval. *Communications of the ACM*, 26(12):1022–1036, December 1983.

[TC90] Howard Turtle and W. Bruce Croft. Inference networks for document retrieval. In Jean-Luc Vidick, editor, *Proceedings of the 13th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 1–24, Brussels, Belgium, September 1990.

[Tur90] Howard R. Turtle. *Inference Networks for Document Retrieval.* PhD thesis, University of Massachusetts, 1990.

[Tur94] Howard Turtle. Natural language vs. boolean query evaluation: A comparison of retrieval performance. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 212–220, Dublin, Ireland, July 1994.

[WK79] W. G. Waller and D. H. Kraft. A mathematical model for a weighted Boolean retrieval system. *Information Processing & Management*, 15(5):235–245, 1979.