Organizing Data for Link Analysis

Ted Senator
DARPA/IPTO
3701 N. Fairfax Dr.
Arlington, VA 22203 USA
Ted.Senator@darpa.mil

John Cheng
Global InfoTek, Inc.
1920 Association Dr., Suite 200
Reston, VA 20191 USA
jcheng@globalinfotek.com

Keywords: Link Analysis, Complex Event Detection, Classification, Intelligence Analysis, Scalability

Abstract

Detection of instances of complex structured patterns in large graphically structured databases is a key task in many applications of data mining in areas as diverse as intelligence analysis, fraud detection, biological structure determination, social network analysis, and viral marketing. Successful pattern detection depends on many factors including the size and structure of the database and of the patterns, the completeness of the available data and patterns, and most important, how the data are divided between analysts performing pattern detection. A combinatorial model based on the metaphor of recognizing and classifying jigsaw puzzles is used to study this problem. Experimental results using this model that yield insights into the effect of various parameters are presented. Alternative data organization strategies are developed, presented, and analyzed. A key result is that the likelihood of puzzle recognition – i.e., pattern detection – depends primarily on the ability to group related data elements in a manner that enables them to be examined by a single analyst or group of analysts.

1. Introduction

A key task in mining linked data is the detection of instances of structured patterns in large databases. (See refs. 1, 4, 5, 7, 9) This task is an essential component of intelligence analysis, biological structure determination, fraud detection, viral marketing, social network discovery, and many other applications of data mining in graphically structured databases. Successful detection of such pattern instances typically occurs through an iterative process of partial matching between pattern specifications and the actual data. The key difficulty is that no piece of information is significant in isolation; rather, it is the combination in context of many related pieces of data that provide indications of significance. Much data are ultimately irrelevant, but this can be determined only after they are connected together. In typical applications such as intelligence analysis, the size of the database, the complexity of the patterns, the large number of partial matches that may or may not be indicative of the patterns of interest, the high degree of incompleteness in the data, and, most important, the combinatorial complexity of considering all possible combinations of data and patterns, make successful pattern detection almost impossible.

Reference 2 presents a mathematical analysis of the combinatorial complexity of link discovery using an abstract model based on a metaphor of classifying jigsaw puzzles in a large collection of pieces sampled from many puzzles. This paper extends that work in several directions. In particular, reference 2 presents results for only a small set of choices of parameter values; this work presents sensitivity analyses with respect to all the parameters of the model. This paper also explores alternative methods of organizing the data and the analytical task and their effects on the likelihood of successful detection.

This paper is organized as follows. First, the model used in reference 2 is described and the key results, many of which are counterintuitive to highly experienced intelligence analysts, are reviewed. It next introduces some model refinements and explores their effects. The largest section of the paper presents and analyzes alternative schemes for organizing the analysis of large volumes of networked data, including pipelining, partitioning, and multi-stage classification, using the basic model as its mechanism for evaluation. The paper ends with conclusions and suggestions for future work.

2.0 The Jigsaw Puzzle Model

This section summarizes the model from reference 2. It describes the metaphor behind the model, presents

the mathematical development of the model, and notes its limitations. The mathematical model is developed based on counting strategies. It is based on the metaphor of being able to classify a jigsaw puzzle as interesting or not based on a large set of pieces sampled from a vast number of puzzles, requiring several pieces from the same puzzle being available to enable recognition, and distributing the sample of pieces among many analysts to handle the workload. For simplicity, we first consider the case of a single analyst, which is equivalent to an automated data mining process that can operate on the entire database at once, and then extend to multiple analysts.

2.1 Metaphor

We imagine that every element of available data is an individual jigsaw puzzle piece with the picture obscured and that pieces from multiple puzzles arrive all mixed together. Recognition of a puzzle (i.e., determination of its significance, modeled as the emergence of the picture) depends on obtaining a minimum number of pieces of the puzzle. Puzzle pieces are assigned randomly and possibly repeatedly to analysts. The model is depicted in Figure 1. The model is used to answer the following questions:

- What is the probability that a person can solve a puzzle of interest (i.e., obtain enough pieces to recognize a particular picture)?
- How does the solution probability depend on various parameters such as the number and workload of analysts, the number of puzzles and of pieces per puzzle, the number of pieces required to recognize a puzzle, and the number of interesting puzzles?
- If analysts collaborate in teams, how does the solution probability change?

More formally, the model assumes that during a specified unit of time, there are **N** puzzles of size **P**, for a total of **NP** pieces. Of these **N** puzzles, **I** are of interest, and **N-I** are not. **S** pieces are examined independently by each of **A** analysts. Recognizing a puzzle requires a minimum of **M** pieces of that particular puzzle. (There are obvious constraints between these parameters required for a sensible and useful interpretation, e.g., **I<N**, **M<S**, etc.) Model parameters are summarized in Table 1.

In this puzzle model, each piece represents an element of linked data. The context-free nature of linked data is captured by the fact that there is no way of distinguishing the puzzle pieces *a priori* and that single puzzle pieces are not meaningful – only a "critical mass" of **M** related pieces enables the analyst to recognize the puzzle. The ratio **M/P** captures the difficulty of detecting a pattern; the inverse may be thought of as the "pattern quality.") The low signal-noise ratio – resulting from much captured data's arising from uninteresting activities – is represented by the model's characterization of interesting vs. non-interesting puzzles. The signal (i.e., data of interest) corresponds to the **IP** interesting pieces; the noise to the **NP-IP** pieces of the uninteresting puzzles. Analyst productivity is modeled by the fact that the analyst only sees **S** puzzle pieces.

Simplifying assumptions of the model are:

- Puzzle sizes are identical
- Information content of all pieces is identical – ignores data quality issues
- No additional structure besides pieces and puzzles
- Probability of assignment of a piece to an analyst is random and uniformly distributed
- Each piece is relevant to only a single puzzle
- No redundant pieces received by an individual analyst – we use sampling without replacement
- No "contradictory" pieces – corresponds to ignoring data inconsistencies
- Pieces are analyzed as a group; i.e., a batch process. Hence, issues such as data distribution over time and decay of memory are ignored.
- Explicit modeling of the time or effort required to recognize puzzles is ignored.

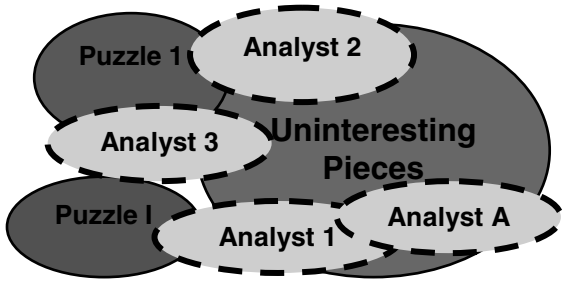


Figure 1

Table 1: Model Parameters

N:	number of puzzles
P:	pieces per puzzle
I:	number of interesting puzzles
A:	number of analysts
S:	pieces per analyst
M:	puzzle recognition threshold

2.2 Mathematical Derivation

The model is analyzed using counting arguments. We first consider the simplest situation in which there is only one puzzle-solver (**A = 1**), and only one puzzle of interest (**I = 1**). What is the probability that the analyst finds a solution?

First, all possible ways in which **S** unique pieces can be sampled from the total number of pieces is:

$$\binom{NP}{S} \quad \text{where} \quad \binom{x}{y} = \frac{x!}{y!(x-y)!} \qquad \text{(Eq. 1)}$$

Using the fundamental counting principle [3], the number of distinct ways that **M** pieces from the single interesting puzzle can be chosen from all pieces is:

$$\binom{P}{M}\binom{NP-P}{S-M} \qquad \text{(Eq. 2)}$$

The first term is the number of ways **M** pieces from the interesting puzzle can be chosen from its set of **P** pieces. The second gives the number of ways the remainder of puzzle pieces – the non-interesting pieces – can be chosen, since the number of pieces in that set is **NP-P**, and the number of pieces picked from that set is **S-M**.

The puzzle can be solved, however, whenever *at least* **M** pieces from the puzzle of interest are drawn. The number of ways this can happen is

$$\binom{P}{M}\binom{NP-P}{S-M} + \binom{P}{M+1}\binom{NP-P}{S-M-1} + \ldots + \binom{P}{S} \qquad \text{(Eq. 3)}$$

assuming that **M < S < P**. The first term in the sum is identical to Equation 2. Each succeeding i^{th} term gives the number of ways that **M + *i*** pieces from the interesting puzzle can be chosen from all pieces.

The solution probability for **A = I = 1** is then:

$$\frac{\sum_{i=M}^{\min(S,P)} \binom{P}{i}\binom{NP-P}{S-i}}{\binom{NP}{S}} \qquad \text{(Eq. 4)}$$

Next we solve the more general case, allowing the puzzles of interest, **I,** to range inclusively from 1 to the total number of puzzles, **N**. We continue to assume, however, that the number of analysts is 1 (**A** = 1). To simplify the mathematics, we compute the complement of the solution probability, i.e., the probability that no puzzles can be solved, and then subtract this from 1 to get the actual solution probability.

The ways that the number pieces per analyst (**S**) can be chosen such that no puzzle of interest can be solved is:

$$\sum_{X_1=0}^{M-1} \cdots \sum_{X_I=0}^{M-1} \binom{P}{X_1} \cdots \binom{P}{X_I} \binom{NP - IP}{S - \sum_{j=1}^{I} X_j} \qquad \text{(Eq. 5)}$$

$$\textit{where} \quad S \geq \sum_{j=1}^{I} X_j$$

Here, each term in the sum gives the number of ways that no puzzles of interest can be solved for a unique combination of pieces per puzzle of interest – i.e., fewer than **M** pieces of any puzzle of interest are present. The **I** nested sums give all possible combinations for which this can be true, resulting in the total number of ways that no puzzle of interest can be solved.

The probability that the analyst cannot solve any puzzle of interest is then:

$$P_F = \frac{1}{\binom{NP}{S}} \infty \sum_{X_1=0}^{M-1} \cdots \sum_{X_I=0}^{M-1} \binom{P}{X_1} \cdots \binom{P}{X_I} \binom{NP - IP}{S - \sum_{j=1}^{I} X_j} \qquad \text{(Eq. 6)}$$

$$\textit{where} \quad S \geq \sum_{j=1}^{I} X_j$$

Hence, the solution probability – i.e. the probability that the analyst solves at least one puzzle of interest – is:

$$P_S = 1 - P_F \qquad \text{(Eq. 7)}$$

Although Equation 7 gives a closed-form solution, its form is such that the solution behavior is difficult to intuit. Hence, a series of computational experiments were conducted to analyze the shape of the solution space.

3.0 Previous Experiments and Discussion

This section reviews experimental results from reference 2.

In the following experiments, a range of parameter values is considered (Fig. 2). They were chosen to roughly match real-world intelligence data characteristics. Analysts can examine about 200 messages/day. That number is used as a basis for **S**, which ranges from 200 to 5000, reflecting the number of "puzzle pieces" an analyst might see in a day,

a week, and a month. Message traffic volume is approximately 10,000 messages/day. Assuming that a reasonable number of puzzles is 20 for small problems (and increasing that by a factor of 5 and 25 for medium and large problems, continuing the day/week/month idea), the number of pieces per puzzle, **P**, must be fixed at 10,000/20 = 500. Finally, the puzzle recognition threshold is chosen at 25, which is 5% of the pieces of each puzzle, a fairly conservative estimate.

Problem Size ->	Small	Medium	Large
N (total puzzles)	20	100	500
P (pieces per puzzle)	500	500	500
S (pieces per analyst)	200	1000	5000
M (recognition threshold)	25	25	25

Fig. 2. Parameter ranges considered in experiments

3.1 Methodology

The following methodology was used:

- Java code was written to compute the probabilities. BigInteger/BigDecimal datatypes were used to ensure sufficient accuracy in the computation.
- Windows XP (1.4 GHz Intel Pentium 4 Mobile CPU, 256MB RAM) was used to execute the code.
- Wolfram Research's Mathematica 4.0 were used to graph the results. Some of the axes are labeled oddly – e.g., "**Number of Puzzles – 75**" indicates that the axis' scale range from 0 to 50 represents an actual **Number of Puzzles** from 75 to 125.

3.2 Puzzle Solution Probability vs. Noise

Figure 3 depicts how the solution probability $\mathbf{P_S}$ varies as the number of puzzles **N** increases while the other parameters are held constant, essentially displaying $\mathbf{P_S}$ as noise increases. Note that in this graph, $\mathbf{P_S}$ is shown on a logarithmic scale.

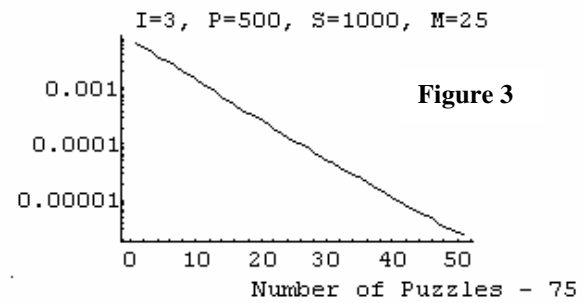


Figure 3

This experiment demonstrates how quickly $\mathbf{P_S}$ falls as noise increases. It suggests that collecting more data that do not contain the phenomena of interest will do more than simply obscure interesting patterns; it will break them apart into pieces too small to enable recognition.

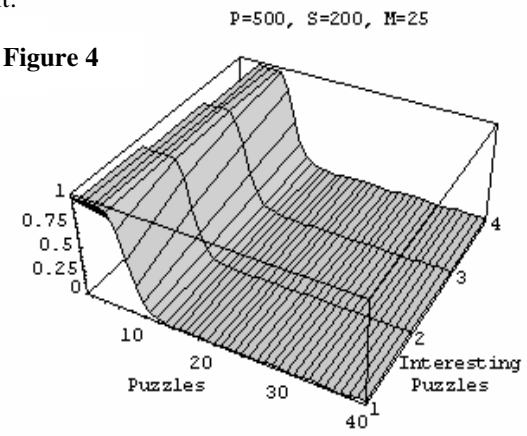
3.3 Signal vs. Noise

In Figure 4, the total number of puzzles is varied along one dimension while the number of interesting puzzles is varied along another, showing how the solution probability $\mathbf{P_S}$ is affected by both these parameters.

The graph shows that increasing the number of interesting puzzles **I** tends to increase $\mathbf{P_S}$; however, this gain seems fairly slow. In the other dimension, we once again see how quickly $\mathbf{P_S}$ falls with increasing noise.

Since increasing signal and noise clearly have opposite effects on $\mathbf{P_S}$, it is interesting to consider their relative effects on $\mathbf{P_S}$. Due to the limited range of **I** in Figure 4, this tradeoff is difficult to see. Figure 5 addresses this issue by plotting $\mathbf{P_S}$ against the number of puzzles when the signal-to-noise ratio is constant.

Figure 4



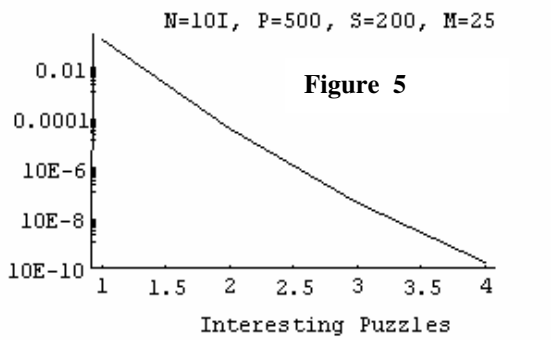


Figure 5

$\mathbf{P_S}$ is plotted on a logarithmic scale; the graphs show that even if **I** varies linearly with **N** (i.e., constant signal/noise), $\mathbf{P_S}$ falls extremely quickly. The prior set of experiments (Figures 3-4) showed that the solution probability falls exponentially whenever extraneous data is introduced, given a *fixed* set of inter-

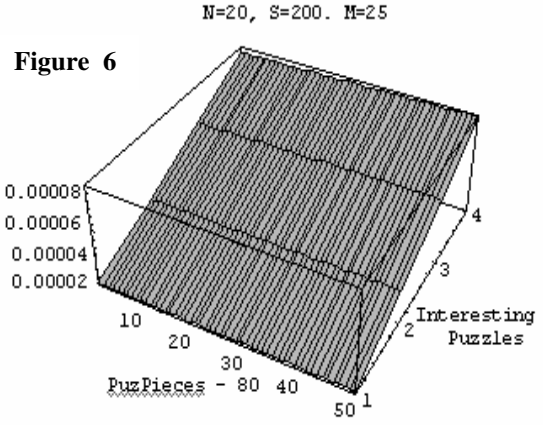
esting data. This experiment demonstrates the stronger result that this exponential decrease in link-discovery efficacy results as the data volume increases *even when the proportion of interesting data remains constant*.

3.4 Signal vs. Number of Pieces per Puzzle

To what extent does the number of pieces per puzzle **P** influence $\mathbf{P_S}$? If the recognition threshold **M** remains fixed, increasing **P** effectively reduces the proportion of the puzzle required to solve it, which should increase $\mathbf{P_S}$. Figure 6 graphs $\mathbf{P_S}$ vs. **P** and **I**.

We see that increasing **P** has an almost negligible effect, especially compared to **I**. This result suggests that increasing the amount of data available to analysts, *even if such an increase assumes that the number of interesting and non-interesting puzzles is fixed*, will not result in significant gains in link discovery performance. More data about the same phenomena does not help much.

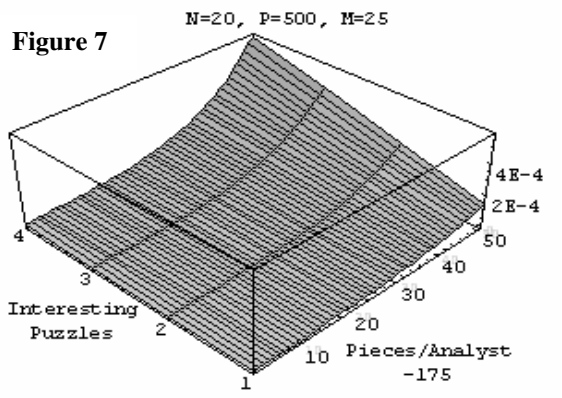
Figure 6



3.5 Signal vs. Pieces/Analyst

As we have seen, $\mathbf{P_S}$ increases with **I**. We would expect the number of pieces per analyst **S** to have a similar effect, since increasing **S** means the analyst is exploiting a larger proportion of the total amount of information, but how do **I** and **S** compare against each other? Figure 7 shows that **S** dominates, causing faster growth in $\mathbf{P_S}$. Increasing **I** while keeping the total number of puzzles constant increases the signal to noise ratio, and essentially amounts to decreasing the amount of irrelevant information available to the analyst – i.e., an increase in data relevance. This experiment shows that in order to increase link discovery effectiveness, increasing the amount of data an analyst can process is more important than increasing data relevance.

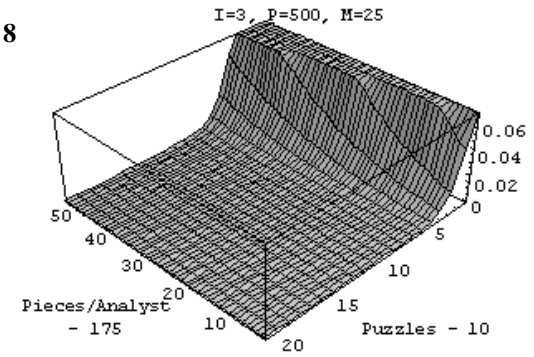
Figure 7



3.6 Noise vs. Pieces/Analyst

Since **S** dominates signal, we now consider the question of **S** compared to noise. Figure 8 shows that although **S** has a considerable affect on $\mathbf{P_S}$, $\mathbf{P_S}$ is much more sensitive to noise. That is, the negative effects of extraneous data on link discovery will always overwhelm any gains that result from improved analyst data cognizance.

Figure 8



3.7 Data Volume vs. Pieces/Analyst

From previous experiments we know that increasing data volume, even with a constant signal-to-noise ratio, decreases $\mathbf{P_S}$. What if the pieces/analyst is increased? Does that offset the poor data-scaling performance of $\mathbf{P_S}$?

Figure 9 shows that $\mathbf{P_S}$ is much more sensitive to data volume than **S**. As in earlier experiments, even when the proportion of relevant and irrelevant data remain fixed, any increase in the amount of available data has a powerful effect on $\mathbf{P_S}$ – significantly more powerful than that of increasing **S**.

3.8 Recognition Threshold vs. Pieces/Analyst

One would expect $\mathbf{P_S}$ to grow rapidly as the recognition threshold M decreases. How does this decrease comparewith an increasing **S**, which (as previously seen) also causes $\mathbf{P_S}$ to grow?

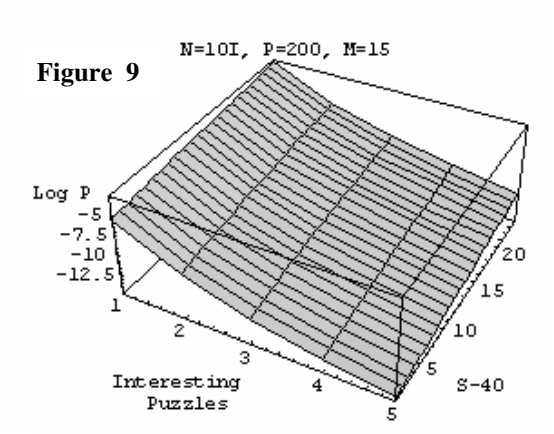


Figure 9

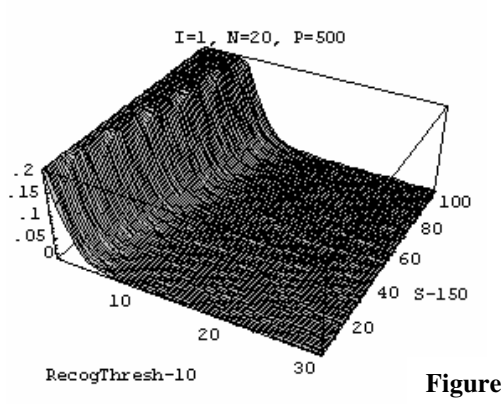


Figure 10

Figure 10 shows that **M** is significantly more important than **S**. This result suggests that efforts to reduce **M** are more valuable than efforts to increase **S** (or efforts to reduce noise, as previously shown). Reducing **M** is somewhat analogous to increasing **S** and the pieces per puzzle, **P**. Hence the dominance of **M** over **S** seems intuitively clear.

Taken together the experiments show that the dominant effect is the critical need to ensure that enough pieces sufficient for recognition will reach an individual analyst. The breaking apart of a puzzle into groups too small for an individual analyst to recognize is what makes link discovery such a difficult analytical task. Conversely, technologies that can reassemble the puzzles, or at least group pieces likely to have come from the same puzzle in a way that enables them to be assigned to the same analyst, should be most valuable.

3.9 Multiple Independent Analysts

Link discovery is difficult for analysts working in isolation because they cannot process enough data. Hence, we now consider the case where multiple analysts attempt to solve the puzzle problem by working in teams. An analysis of independent analysts is conducted. We do not include herein the analysis of collaboration from reference 2 as no additional results depend on that analysis.

The probability that all **A** analysts fail, assuming that they operate independently, is:

$$\prod_{i=1}^{A}(1-P_i)$$

where $\mathbf{P_i}$ equals $\mathbf{P_S}$ for the *i*th analyst. Figure 11 shows how $\mathbf{P_S}$ scales with the number of analysts, assuming $\mathbf{P_i}$ for each analyst is 1%. The function is strictly monotonically increasing, which is a desirable characteristic. Adding manpower always results in higher levels of performance, albeit with diminishing returns.

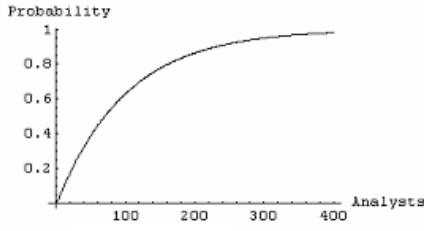


Figure 11

However, it is infeasible to use multiple independent analysts as a link discovery strategy. Consider the following example:

- 10 total puzzles (N = 10)
- I = N/10 (1 puzzle of interest)
- 1000 pieces per puzzle (P = 1000)
- 200 pieces per analyst (S = 200)
- Recognition threshold = P/20 (M = 50)

This represents a relatively "easy" problem; puzzles of interest comprise a high percentage of the total (10%), the recognition threshold is only 5%, and analysts can see 2% of the total amount of data. These parameters give $\mathbf{P_S = 5.53x10^{-10}}$. Employing even 200 million analysts gives only a 10% probability of finding the puzzle of interest!

4.0 More Analysts with Less Data to Analyze Experiments

As more analysts are available the amount of data analyzed by each analyst may change, or the amount of time the analyst has available to examine the same set of data may vary. This section models the variation in analyst productivity that would result from analyzing different amounts of data. The model presented earlier, however, cannot directly model this quantity. Hence, we model productivity increases indirectly by allowing M – the puzzle recognition threshold – to vary as a function of other parameters. Using this modification, decreases to M reflect increases in analyst productivity.

In the following experiment, depicted in figure 12, we assume that M stays constant. This gives a conservative baseline, assuming no "productivity gains" from reduced data sets. We see the Ps decreases quickly as A increases due to data fragmentation.

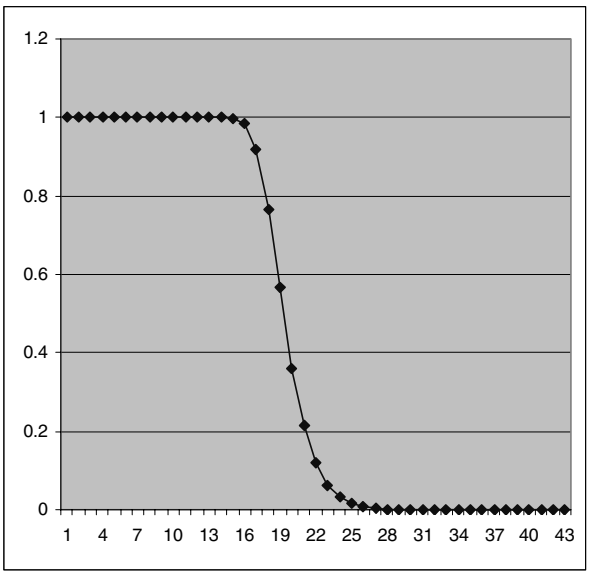


Figure 12

Suppose, however, that analyst productivity increases as data processed by each analyst decreases. How is Ps affected? In the following experiment depicted in figure 13, we model analyst productivity increases by dividing M – the puzzle recognition threshold – by a function that increases linearly with A.

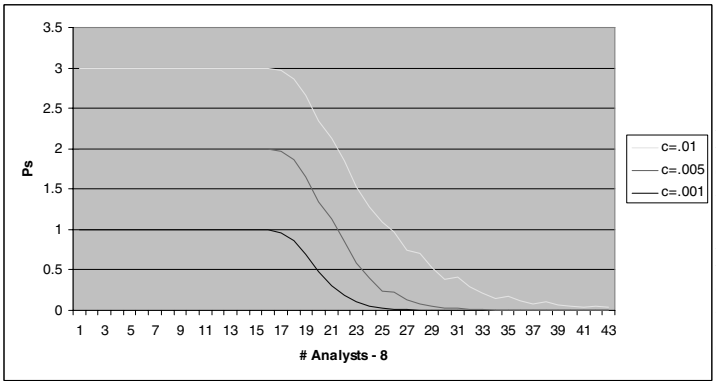


Figure 13

The three curves correspond to different slopes of the linearly increasing function. We see that this "optimistic" model of productivity is outweighed by data fragmentation.

The next experiment (figure 14) uses perhaps the most realistic model of analyst productivity. Here, M is divided by a function that increases sub-linearly with A (in this case, sqrt(A)).

As expected, Ps in this experiment is bracketed by the experiments in figures 11 and 12.

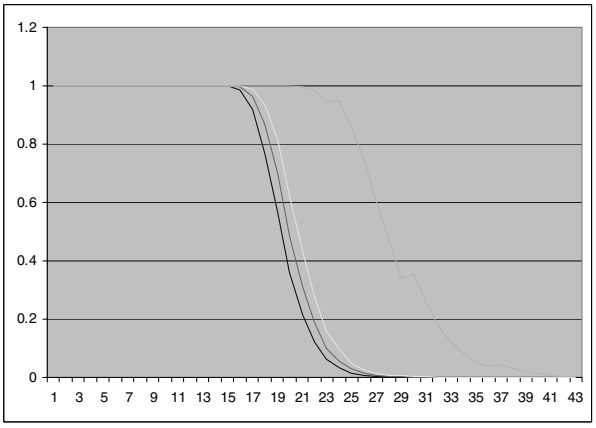


Figure 14

5.0 Data Partitioning

Our puzzle model assumes a single pool of data. However, in real-world organizations, data may be divided among analysts in various ways – geographically, by type, etc. This data segmentation may lead to an interesting phenomenon – it may result in differences in signal-to-noise ratios between the partitions. This leads to an interesting question. Assume that data partitioning does in fact give S/N differentials. How do these differentials affect Ps when s/n for each partition is known? When unknown?

To investigate these effects, we model data partitioning and s/n differentials as follows. Given a set of data, we recursively split it in half. If the overall s/n of the set equals R, we assume that after the split, one partition's s/n equals R+d, and the other's s/n equals R-d, where d is the "partition differential."

By recursively splitting the data in this fashion, we can investigate partitions of size 1, 2, 4, 8, …, with various s/n distributions.

In the figure 15 experiment, we consider Ps under different data partitioning conditions. In particular, we vary the partition differential from 0 to 15%, and consider partitions of sizes 1, 2, 4, 8, 16, 32, 64, 128, and 256. The number of analysts is split evenly between the partitions.

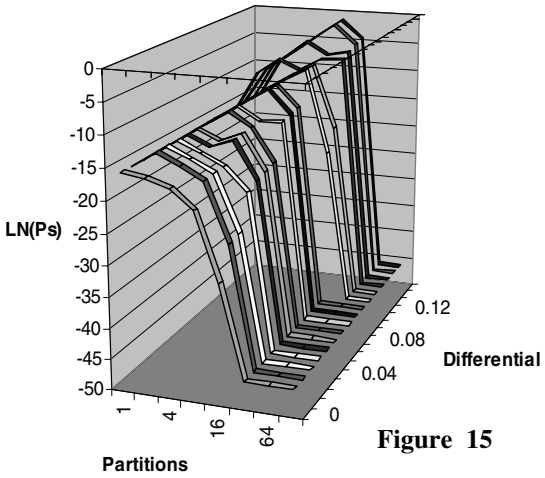


Figure 15

In general, as the number of partitions grows very large, Ps approaches 0 because the resulting data fragmentation prevents success even with extremely high signal/noise. However, this graph shows an interesting interaction between the differential and number of partitions. At times we see a single peak; at other times we see multiple peaks. This suggests a complex interaction between the differential and the number of partitions.

Additionally, this experiment demonstrates that when data partitioning leads to even relatively modest s/n differentials, such partitioning increases Ps *even when s/n for each is unknown*. This clearly has ramifications with respect to data organization in link discovery systems.

The previous experiment assumed only that a s/n differential existed between partitions, but it did not assume that the "better" partitions could be recognized – hence an even split of analysts between partitions. Assume, however, that we could identify the single partition with the highest signal/noise. Assuming that analysts can only work on one partition, one strategy would utilize all analysts on that single partition. How would this effect Ps?

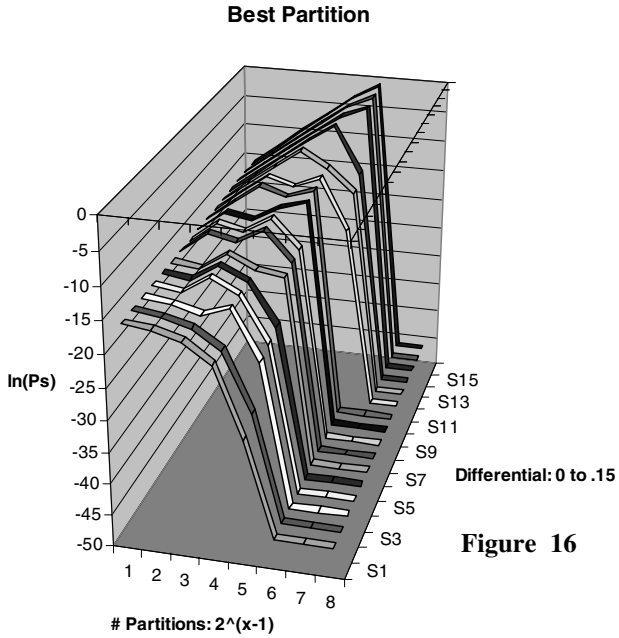


Figure 16

In this experiment (figure 16), we see curves with similar characteristics to those of Figure 15. As expected, Ps for this strategy is uniformly higher here.

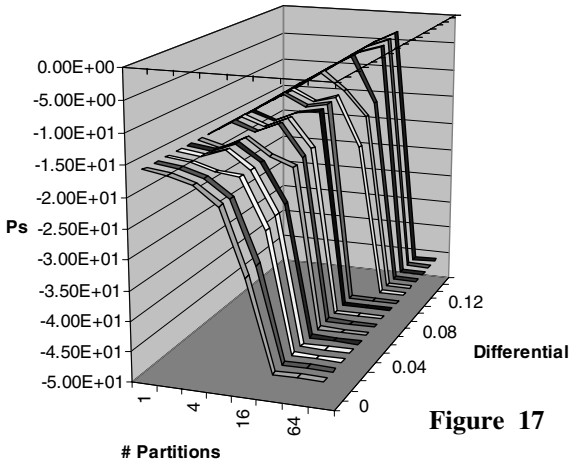


Figure 17

Finally, we consider a strategy whereby the number of analysts assigned to each partition is proportional to its signal-to-noise ratio. This allocation satisfies the well known resource allocation result of microeconomics – that marginal benefit be equal to marginal cost. Once again in figure 17 we see a similar set of curves. Ps values in this experiment lie between those of the prior two experiments. Note that the strategy of partitioning data recursively matches the detection process proposed in reference 8.

6.0 Data Partitioning with Parameter Variation (Sensitivity Analysis)

In the following experiments, we investigate the effects of various parameters on Ps when combined with data partitioning.

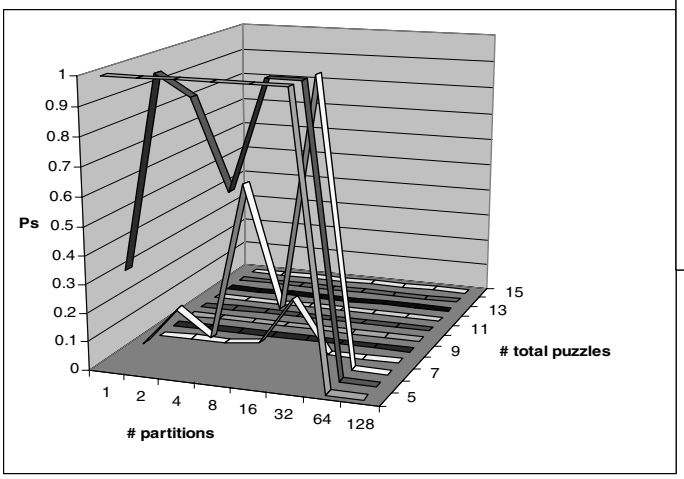


Figure 18

Figure 18 shows how the number of partitions affects Ps as the total number of puzzles increases (with 10% partition differential, I=1, P=1000, S=200, M=50, 256 analysts). As expected, increasing the total number of puzzles reduces Ps. The number of partitions, however, affects Ps in an interesting manner. The varying peaks in Ps suggests the Ps is extremely sensitive to the number of partitions and partition variance.

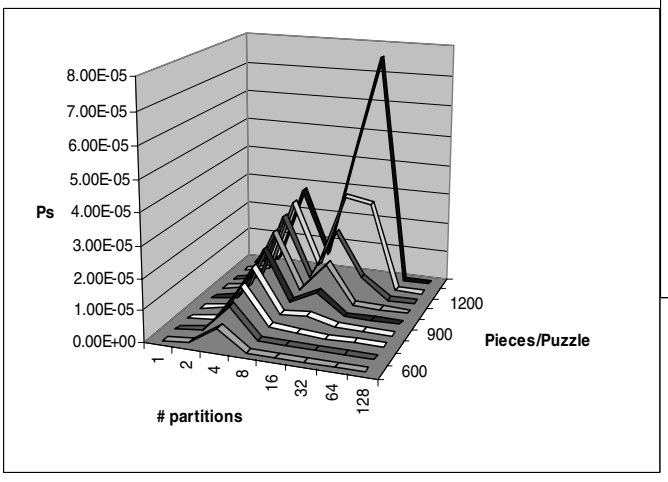


Figure 19

Figure 19 gives Ps as the number of partitions varies with P (with 10% partition differential, I=1, N=10, S=200, P=50, A=256). This family of curves have a two-peak characteristic, one slow-growing and the other much faster growing. It suggests that the number of partitions strongly affects Ps.

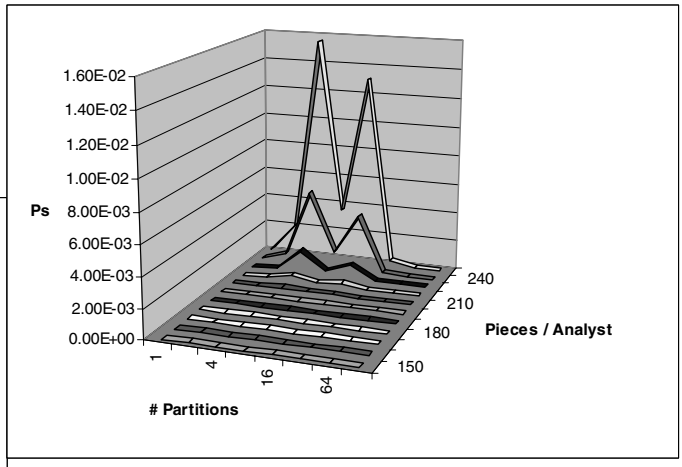


Figure 20

The above figure 20, graphing Ps against the number of partitions and pieces/analyst, displays a two-peak characteristic. This suggests an interaction between the number of partitions and the partition differential.

In figure 21 we see how the recognition threshold, in conjunction with the number of partitions, affects Ps. Once again, we see a two-peak characteristic with these parameters.

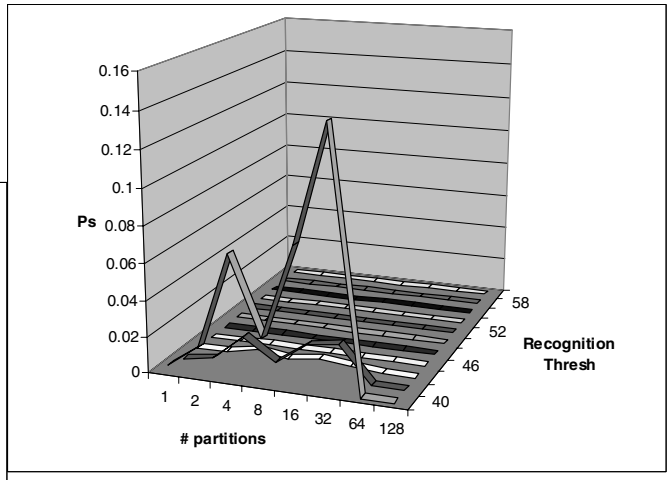


Figure 21

7.0 Data Pipelining

An alternative strategy for analyzing large amounts of data is pipelining. Using this strategy, the analysts would be divided into groups, each of whom would work on a rolling window of data for the time duration of the window. So, for example, the analysts might be divided by day of week, and each group would work on a seven-day window of data for a full seven days until the next week corresponding to their day of the week was reached. This strategy is modeled in comparison with the baseline strat-

egy by dividing A by 7 while leaving other parameters unchanged. The final probabilities from equation (7) are them multiplied by 7 to account for all the groups. As in the data partitioning experiment, the likelihood of pieces' being related is not changed based on how the data are divided.

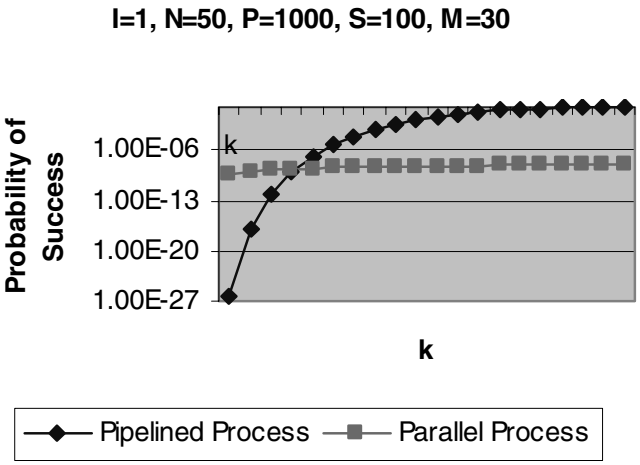


Figure 22

8.0 Conclusions and Future Work

These experiments demonstrate that the ability to detect complex structured patterns in large graphical databases depends not only on the characteristics of the data and the patterns, but also – and even more importantly – on the strategies used to organize the detection process. There are complex interactions between the nature of the analytical process and the characteristics of the data and patterns that can have drastic effects of the ability to detect instances of patterns successfully, with minor changes in characteristics able to cause major changes in the probability of success.

The most important future extension of this work would be to relax the assumption of the equal likelihood of connectivity between puzzle pieces in different partitions or separated by a different temporal extent and examine the effects on pattern detection capability. Explicit modeling of pattern and data structures and connectivity in the context of what is known about how real networks are formed would be both extremely useful and difficult.

9.0 Acknowledgements

The second author's work on this paper was sponsored by the Air Force Research Laboratory, under Contract F30602-01-C-0202. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either express or implied, of the Air Force Research Laboratory, the Defense Advanced Research Projects Agency or the U.S. Government.

10.0 References

[1] Adibi, J., Chalupsky, H., Grobelnik, M., Milic-Frayling, N., and Mladenic, D. (Eds.) *Second International Workshop on Link Analysis and Group Detection (LinkKDD-2004)*. (Seattle, WA, USA, August 22, 2004).

[2] Cheng, J. and Senator, T. A Combinatorial Analysis of Link Discovery. In ***Proceedings of the 2005 International Conference on Intelligence Analysis*** (IA-2005). (McLean, VA, USA, 2-6 May 2005). The MITRE Corporation, available at https://analysis.mitre.org/proceedings/

[3] Deskins, W. E., *Abstract Algebra*, pp. 45-50. Dover Publications, 1996.

[4] Goldberg, H.G., and Senator, T.E. Break Detection Systems. In *AI Approaches to Fraud Detection and Risk Management: Collected Papers from the 1997 Workshop* Technical Report WS-97-07 AAAI Press, Menlo Park, CA.

[5] Jensen, D. and Goldberg, H. Artificial Intelligence and Link Analysis: Papers from the 1998 AAAI Fall Symposium, AAAI Press, Menlo Park, CA 1998

[6] Jensen, D., Rattigan, M., and Blau, H. Information Awareness: A Prospective Technical Assessment. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*. (Washington, DC, USA, August 24-27, 2003). ACM Press, New York, NY, 2003, 378-387.

[7] Mladenic, D., Grobelnik, M., Milic-Frayling, N., Donoho, S., and Dybala, T. (Eds.) *Workshop on Link Analysis for Detecting Complex Behavior (LinkKDD2003)* KDD2003, (Washington, DC, USA, August 2003).

[8] Senator, T.E. Multi-Stage Classification. To appear in *Proceedings of the International Conference on Data Mining 2005 (ICDM 2005)*. (Houston, TX, 27-30 November 2005). IEEE.*

[9] Senator, T. E., and Goldberg, H. Break Detection Systems. In *Handbook of Data Mining and Knowledge Discover*y, W. Klosgen and J. Zytkow (eds.), Oxford University Press. 863-873, 2002.