A Hybrid Bayesian Network Based Multi-Agent System and A Distributed Systems Architecture for the Drug Crime Knowledge Management*

Parag C. Pendharkar  
Associate Professor, Information Systems  
School of Business Administration  
Pennsylvania State University at Harrisburg  
777. W. Harrisburg Pike  
Middletown, PA 17057  
Phone: (717)-948-6028  
Fax: (717)-948-6456  
e-mail: PxP19@psu.edu

Rahul Bhaskar  
Associate Professor, Information Systems  
School of Business, CSU-Fullerton, 800  
California State University at Fullerton  
North College Blvd., Fullerton, CA 92831.  
e-mail: rbhaskar@Exchange.FULLERTON.EDU

Bahador Ghahramani  
Associate Professor, Information Systems and Quantitative Analysis  
College of Information Science and Technology  
University of Nebraska at Omaha  
Omaha, Nebraska 68182-0116 (USA)  
Tel:(402) 554-3975, Fax: (402) 554-3400  
E-mail: bghahramani@mail.unomaha.edu

Key Words: Distributed Artificial Intelligence, Intelligent Agents, Experimental Design, Distributed Computing, Object-Oriented Programming

Submitted to: Journal of Information Technology and Decision Making

Written: January 4, 2003  
Revised: July 12, 2003  
Please don’t quote without permission from the authors

* Some of the components described in this paper are proprietary and may be available from Sherpa Analytics Inc. (http://www.sherpa.biz).
A Hybrid Bayesian Network Based Multi-Agent System and A Distributed Systems Architecture for the Drug Crime Knowledge Management

ABSTRACT

In this paper, we describe an approach for building a hybrid bayesian network based multi-agent system for drug crime knowledge management. We use distributed artificial intelligence architecture to create a multi-agent information system that integrates distributed knowledge sources and information to aid decision-making. Our comparison of the hybrid system with a previously developed stand-alone expert system Sherpa, which was in use at a large drug crime investigation facility, shows that the current system compares similar to the existing system in terms of efficiency and effectiveness of knowledge management. We illustrate how the proposed hybrid bayesian network-based can be implemented in the distributed computing network environment.

Key words: Distributed Artificial Intelligence, Multi-Agent Information Systems, Distributed Problem Solving, Distributed Computing
1. Introduction

Distributed Problem Solving (DPS) is an approach to solve problems that can be decomposed into sub-problems which can be solved by independent problem solving agents implemented as loosely coupled modules (also called agents in multi-agent systems). These modules (agents) cooperate and share knowledge in the form of tasks or data about a problem to develop a solution\textsuperscript{11,18}. Many real world situations can be modeled as a set of independent cooperating agents\textsuperscript{17,12,20}. The need for modeling problems as a set of independent cooperating agents is characterized by an increasing complexity of problems, which cross-functional boundaries, and an increasing trend towards specialization of skills in narrow functional areas\textsuperscript{16,14}.

In this paper, we detail our experience from the design and implementation of a hybrid bayesian network-based DPS multi-agent information system (MAIS). Our problem is that of drug crime investigation in the state of Wisconsin. The drug investigation problem includes various steps such as target (suspect identification) selection, data collection, data evaluation and analysis and dissemination of information and data. We focus on the data evaluation, management and analysis part of drug investigation problem and implement the hybrid system as a knowledge management and decision support tool. We evaluate the performance of the proposed hybrid system with the existing stand-alone system called Sherpa\textsuperscript{2}. Additionally, we describe a distributed systems architecture for implementing the proposed system in a distributed computing-based client/server environment.
The proposed distributed systems architecture provides a web-based deployment, which makes it independent of any particular hardware. The stand-alone system Sherpa is not web-enabled and does not provide global access for the law enforcement agents. The new web-based distributed systems architecture provides an internet based global access, and scalability that is necessary for a modern day law enforcement systems. The new system and its components are proprietary, and our paper describes these components at a higher level. Some of the components described in our paper may be available at Sherpa Analytics Inc. (http://www.sherpa.biz).

The rest of the paper is organized as follows: in Section 2, we provide an introduction of distributed artificial intelligence (DAI) and distributed problem solving. In Section 3, we describe the traditional drug crime investigation process at Wisconsin Division of Narcotics Enforcement (WDNE). In Section 4, we describe the existing system Sherpa and its environment. In Section 5, we propose a modification to the existing system and propose a modified bayesian-network based system. In section 6, we propose a distributed systems architecture for the implementation of the proposed system. In Section 7, we conclude the paper with highlighting implications of the current research.

2. Distributed Artificial Intelligence and Distributed Problem Solving

A distributed artificial intelligence (DAI) system is composed of a set of separate modules (also called agents if each module acts on a part of the problem autonomously) and a set of communication paths between them. Distributed problem solving (DPS) is a sub-domain of distributed artificial intelligence where the agents/modules in DAI are decentralized and loosely coupled knowledge sources (KSs) or problem solvers. The
KSs cooperate in the sense that none of them has sufficient information to solve the entire problem itself. The KSs have to work as a team and share knowledge about the tasks, results, goals, and constraints to solve the problem. The term “decentralized” in DPS context means that both control and data about a problem are logically and geographically distributed (i.e., there is no global data storage), and the term “loosely coupled” means that individual KSs spend most of their time in computation rather than communication\textsuperscript{18,12}.

In a distributed problem-solving process, a problem is divided into tasks, and special task performers (agents) are designed to solve these tasks for the specific problem. All the interaction strategies between the agents are incorporated as an integral part in the design of the system (also called multi-agent system). Most problem solving processes in a DPS system consists of four phases. These are as follows\textsuperscript{15}:

1. The decomposition of the problem into sub-problem tasks;
2. The allocation of the sub-problem tasks among the agents;
3. The solving of the sub-problem tasks among the agents;
4. The integration of the solutions, obtained in previous phase, to obtain the global solution.

A DPS system has significant advantages over a single, monolithic, centralized problem solver. These advantages include: 1) Faster problem-solving by exploiting parallelism, 2) reducing communication by transmitting only high-level partial solutions rather than raw data to a central site, 3) increased flexibility by creating problem solvers with different abilities dynamically combined, to solve the current problem, and 4) increased reliability by allowing other problem solvers to replace failed ones\textsuperscript{6,19}. 
3. The Traditional Drug Crime Investigation Process in Wisconsin

The drug crime investigation is a process consisting of a series of highly interrelated components. A failure or weakness in any one of these components seriously impairs the entire process and reduces the quality of product. The various components of a drug investigation process are shown in Figure 1. Among the steps in the drug investigation process are:

- **Target (suspect) Selection** - The systematic selection of targets (suspects) to insure that intelligence efforts are directed towards targets which are in acceptable balance of utility (worth), probability of a successful result, and resources expended.

- **Collection** - Collection consists of both data collection planning and actual collection of information regarding the target(s) of intelligence operations. The collection must first be carefully planned before any information gathering occurs. Information is collected from both overt (open) and covert (closed) sources.

- **Data Evaluation** - All collected information cannot be validated as factual. Intelligence may be facts, opinions, rumors, and/or inferences. Many times one piece of intelligence may contradict another. It is imperative that each intelligence report include both an evaluation of the source’s reliability and the reported information’s validity.

- **Storage and Retrieval System** - Collected intelligence must be promptly recorded on the appropriate intelligence form and placed in a storage and retrieval system. The system must include:
  - rapid user access,
  - selectivity of retrieval,
- documentation of each dissemination,
- periodic system audit and, when appropriate, information purging,
- physical security to protect the files, and
- system security to protect the information flow.

- Data Collation - Data collation is defined as “assembling data in proper order to clarify or give meaning to information.” Taking a deck of playing cards and arranging them into four piles, one of each suit, in numerical order is one example of collation as is arranging a stack of surveillance reports in chronological order so that the oldest report is on top and newest is on the bottom before reading them. Many techniques used by the intelligence analysts such as link charts, various flow charts, and visual investigative analysis are in fact data collation techniques used in what is commonly referred to as the data description and integration phase of the intelligence process.

- Analysis - Analysis is the heart of an intelligence system. Analysis is a sub-system within an overall intelligence system. The analysis stage involves:
  - data integration and clarification;
  - inference development;
  - inference testing;
  - finalizing inferences which are relevant and meaningful to the user.

- Dissemination - Dissemination is where the entire information is offered to the agency’s officers in usable form upon request.
The traditional drug crime investigation process in Wisconsin consists of data collection from 30 human agents across the state, generating monthly reports, identifying suspects, collecting information about the suspects, and analyzing data leading to the arrest of the suspect. The traditional process used by division of narcotic enforcement (DNE) is very cumbersome. This is due to the combined effects of various factors. The current system includes the fact that only two analysts serve thirty human agents. The human agents lack the data analysis tools. The communication between analysts (situated in Madison) and the DNE human agents (situated in Madison, Appleton, Eau Claire and Milwaukee) is via telephones, e-mails and faxes. The human agents fill out reports about suspects and mail it to the central division in Madison where these reports are entered in a database. From the central database, statisticians analyze the data. The data quality is checked and further analyzed by the supervisors (analysts). The two analysts, one of
which is a senior analyst and the other director, check for format, integrity of data elements, consistency, and accuracy of each data element. After analyzing the data, notes are prepared and are forwarded to a secretary who types the final report.

One of the elements in the final report is the list of suspects. Once the suspects are identified, data is gathered regarding their activities and background. Some of the data is obtained by police undercover and surveillance operations and other data is obtained from external sources such as, Internal Revenue Services and Division of Criminal Investigation. After data about a suspect is obtained, it is analyzed and the suspect is arrested. After the arrest of the criminal, the results of the data analysis are used against the suspect to determine the level of charges against the suspect.

There are several problems related to the traditional drug crime process. First, there is a security and control problem. The information acquired by 30 agents changes many hands for accuracy checks before final report is created. Since each person involved in accuracy check focuses only on the part of the report and not on the entire report, sensitive information is passed from one person to the other. Second, the current process has many redundancies, and inefficiencies. One of the inefficiency is related to the sequential nature of the process. Since the report goes from one person to the other only one person can work on the report at a time. Further, if errors are identified, at least two people are involved in updating the information (one is the supervisor and the other is data entry clerk). The data entry clerk, upon verification of the supervisor, corrects the errors.
4. Description of Sherpa and its Environment

The Sherpa was developed in conjunction with a major reengineering and IT deployment effort undertaken with the IBM. Among the joint deliverables of the project were a set of redesigned business processes, a set of distributed computer systems that allowed the human agents to directly input their reports into the computer, a centralized drug enforcement database and a decision support tool called Sherpa.

In the new reengineered system, human agents directly input their reports into the computer, and monthly report is generated directly by senior management using a user friendly Executive Information System. Figure 2 and Figure 3 illustrate one example of the traditional and reengineered process of creating monthly report. Texas instrument product Information Engineering Facility (IEF) was used for redesigning the business processes.

![Flowchart of Report Generation Process]

**Figure 2: Monthly Report Generation using Traditional Approach**
The new reengineered approach maintains security because only the authorized staff is allowed to check for accuracy. Unlike traditional approach where all the information about a suspect was passed from one person to the other in a sequential manner, the reengineered approach limits the access of information based on the decision making need. In comparison with the traditional approach, the reengineered approach allows for parallel decision-making as everyone can have access to the information at the same time. Reengineered approach also reduced some of the redundant functionality (such as typing the report) in the traditional approach. Among other benefits, reengineered approach made development and usage of advanced decision support tools such as the Sherpa feasible.

The drug crime investigation process requires data acquisition from different sources. For the identified suspects, information is acquired about them. For example, if a suspect is identified then his/her telephone records and bank transactions are checked.
Other information such as surveillance and suspect discarded trash analysis (to check for possibility of traces of drugs in the trash discarded by suspect) is gathered. The different data sources used for data analysis problem are shown in Figure 4. The data sources are:
1) Financial data; 2) Toll Record; 3) Surveillance data; and 4) Other Sources (e.g., trash check and mail check).

Given the distributed nature of data acquisition and the different type of analysis required on the acquired data, DPS approach was well suited to develop the Sherpa. Figure 4 illustrates the dependency diagram for classifying a suspect using the Sherpa. The overall problem of classifying a suspect can be broken down into 4 sub-problems, which need different data sources. Each of these sub-problems can be analyzed independently using different knowledge sources and techniques.

![Figure 4: Dependency Diagram](image-url)
The solutions of the sub-problems can be finally combined to correctly classify the suspect. A simple *bagging* algorithm, illustrated in Figure 5, can be used to correctly classify the suspect. The sources can also be used independently to classify a suspect. The sources may also serve to confirm each other’s results. An example of analysis conducted independently can be a telephone record analysis that is used to determine the relationship of the suspect with a person with a known criminal record. This is possible by observing the calls made by the suspect. If a call is made to a person who has a criminal record then the suspicion about the caller increases. An example of sources working together would be a suspect with a high phone bill, which elicits the need to determine sources of income (financial analysis).

<table>
<thead>
<tr>
<th>For each of 4 models:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predict class of instance using model.</td>
</tr>
<tr>
<td>Return class that has been predicted most often.</td>
</tr>
</tbody>
</table>

**Figure 5: The Bagging Algorithm for final Classification of the Suspects**

The Sherpa uses three intelligent agents as shown in Figure 6. An intelligent agent contains a database for previous cases and a rule based knowledge source based on an inductive, statistical or knowledge engineering approach. The induction learning technique used is Quinlan’s C4.5. In a recent study by Bhattacharyya and Pendharkar⁴, C4.5 is one of the more accurate classification technique compared to non-linear neural networks and genetic programming, and linear genetic algorithms and discriminant analysis based classification methods over a wide range of non-parametric data distribution characteristics. For knowledge engineering agent, protocol analysis and
process tracing techniques were used for knowledge acquisition. Specifically, the
approaches suggested by Grabowski\textsuperscript{8}, and Hoffman\textsuperscript{9} were used for knowledge elicitation.
Probabilistic and discriminant models were used for classification of suspect using
statistical methods. Please see appendix for a brief introduction on these techniques.

The \textit{Sherpa} consists of 5 different modules. These modules are the language
system, the problem processing system, meta-level knowledge, knowledge sources and
databases. Figure 6 shows the architecture of the \textit{Sherpa}. The language system
represents the human-computer interaction module of the \textit{Sherpa}. The language system
is a graphical-user interface module that is used to input data and gain output information
from the \textit{Sherpa}.

The problem processing system is the module by which the past classifying
information on any data is compared to the classifying information being analyzed in the
present problem. Using the Level 5 Object inference engine, any information that pre-
exists in the system is combined with the new information and made available to the
agent. For example, in the telephone record analysis, the telephone numbers of the
suspect are matched with the pre-existing criminal records and made available to the
investigator. This provides the user with more valuable information. The Department of
Justice Secure Wide Area Network and Local Area Network is used to access
information from relevant knowledge sources.
The meta-level knowledge system provides “knowledge about knowledge.” It determines the priority of different knowledge sources in solving the problem. The Sherpa lets the agent choose the desired analysis. Using the meta-level knowledge rules, the Level 5 Object inference engine makes the correct information available to the agent for decision-making. Knowledge sources are independent sources to extract information from data available to the Sherpa. These sources are independent in that they don’t invoke one another and ordinarily have no knowledge of each other’s expertise or
behavior. They are also cooperative in that they contribute solution elements to a shared
problem. Two knowledge sources were developed. These were the telephone and
financial analyses knowledge sources. Databases feed data to the intelligent agents in the
Sherpa. Logically and geographically distributed data is available to the intelligent agents
for decision-making.

5. Hybrid Bayesian-Network Based Multi-Agent System

The Sherpa, due to its dependency on Level 5 Object expert system shell, is hard
to deploy over the internet. Additionally, the meta-level knowledge makes Sherpa too
specific for Wisconsin DNE. One way to enhance Sherpa, so that it can be deployed
over the internet, is to avoid its dependency on static rules and Level 5 Object expert
system software. We replace the meta-level knowledge with a Bayesian network based
classifier. Bayesian network based classified provides the desired flexibility and
adaptability for the Sherpa. Bayesian network can continuously adapt its probabilities
and will have low maintenance cost, when compared to static rule based systems. The
Bayesian network architecture allows the needed internet deployment of the Sherpa.

Bayesian network is a directed acyclic graph that encodes a joint probability
distribution over a set of random variables $\mathcal{R}$. Specifically, a Bayesian network for $\mathcal{R}$ is
set of ordered pair $B = (G, \xi)$, where $G$ is the directed graph and $\xi$ is the set of parameters
that quantifies the network. The vertices of the graph $G$ correspond to the random
variables $X_1, \ldots, X_n$ and the edges represent the direct dependencies of the variables. The
graph encodes independence assumption and is acyclic. In case of drug crime analysis,
let $\mathcal{R} = \{\text{response #1, response#2, response#3}\}$, where response#1, response#2, and
response\#3 are individual responses from intelligent agents \#1, \#2, and \#3 respectively (see Figure 6 for these agents). A response from an agent consists of the agent’s classification of the suspect. Figure 7 illustrates the graph structure of the proposed hybrid system. If the Suspect status is determined by values of broker, customer, source, pursue and non-suspicious then using bayes rule the overall belief in the suspect classification can be written as:

\[
P(Suspect = i \mid response\#1, response\#2, response\#3) = \alpha P(Suspect = i) \prod_{k=1}^{3} P(response\#k \mid Suspect = i).
\]

Where \(i = \{broker, customer, source, pursue, non-suspicious\}\) and \(\alpha\), a normalizing constant defined as \(\alpha = [P(response\#1, response\#2, response\#3)]^{-1}\).

---

**Figure 7: The Hybrid Bayesian Network-Based System**
For a given case, responses from the agents are taken and conditional probabilities for all the possible suspect classification are calculated. The case is classified as based on the conditional probability that receives the maximum probability.

Let $IV = \{ \text{response#1}, \text{response#2}, \text{response#3} \}$ be the set of independent variables with its elements represented as $iv_1 =$ agent#1 response taking one of the five possible values from its value set of $IVValue_1 = \{ \text{broker, customer, source, pursue, non-suspicious} \}$, $iv_2 =$ agent#1 response taking one of the five possible values from its value set of $IVValue_2 = \{ \text{broker, customer, source, pursue, non-suspicious} \}$, and $iv_3 =$ agent#3 response taking one of the two possible values of its value set $IVValue_3 = \{ \text{broker, customer, source, pursue, non-suspicious} \}$. Let $DV = \{ \text{Bayesian Classification} \}$ be the dependent variable taking values from its value set $DVValue = \{ \text{broker, customer, source, pursue, non-suspicious} \}$. Further, let $Examples = \{ \text{examples}_1, \ldots, \text{examples}_n \}$ be the set of all the $n$ historical examples in the database. Each element of the set of $Examples$ consist of one example which can be represented as union of all the elements of the set of $IV$ and $DV$ with each element taking a certain allowable value from its value set. For example, an $element_i$ for some $i \in \{1, \ldots, n\}$ may be as follows.

$$element_i = \{ \text{agent#1=non-suspicious, agent#2= pursue, agent#3=pursue} \}.$$  

We use the following procedure, illustrated in Figure 8, to learn the prior and conditional probabilities of the Bayesian network classifier.

Using the performance measure of identification frequency, we compare the proposed system with the Sherpa. Identification Frequency is the number of drug dealers that are correctly identified as belonging to middle and upper level. The upper level categories consisted of broker and customer, and the middle level category consisted of source.
Learn_Bayes_Network-Classification (Examples)

Assuming $dv_1$ implies $DV=\text{non-suspicious}$, $dv_2$ implies $DV=\text{pursue}$, $dv_3$ implies $DV=\text{source}$, $dv_4$ implies $DV=\text{customer}$, and $dv_5$ implies $DV=\text{broker}$ and assuming $|dv_k|$ means the sub-set of Examples for which the value of $DV=dv_k$, where $k \in \{1,2,3,4,5\}$, we can use the following procedure to obtain prior and conditional probabilities for the naïve Bayes network.

For each value of $dv_k$

$$P(dv_k) \leftarrow \frac{|dv_k|}{|\text{Examples}|}$$

For each $iv_j$ and each of its values in the value set $IVValue_j \ \forall \ j \in \{1,2,3\}$

$$P(iv_j = IVValue_j \ | \ dv_k) = \frac{P(iv_j = IVValue_j \cap dv_k)}{P(dv_k)}$$

Figure 8: The naïve bayes procedure for learning prior and conditional probabilities

Middle and upper level drug dealers can be defined as the drug dealers who are not ‘street pushers’. Their identification is one of the main purposes of drug crime investigation. We use the data from our previous study and simulate a few cases based on the original data. The simulated cases consisted of taking the ignored cases with missing information and completing the missing information with the best possible value; determined by comparing the case with the missing information with its closest matching
case with the complete information. Overall, we had 50 test cases. We randomly divided these 50 cases into 10 sets of 5 cases each. For each set of 5 cases, using the Sherpa and proposed Bayesian network-based system, the number of identified middle to upper level drug dealers were recorded. A t-test for difference of means between Sherpa and bayesian network-based system (BNS) was performed. Table 1 illustrates the results of the t-test.

Table 1: Results of the Paired Sample t-test on Difference of Means

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Differences of Mean</th>
<th>t-value</th>
<th>Df</th>
<th>2-tail sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sherpa vs. BNS</td>
<td>0.1</td>
<td>-0.36</td>
<td>9</td>
<td>0.726</td>
</tr>
</tbody>
</table>

The results of the difference in means between the Sherpa and BNS are not significant. This indicates that BNS performs as well as the Sherpa. At first, it might appear that if no difference between the Sherpa and BNS is seen then why consider an alternative approach? The advantage of BNS is that it can update its probabilities as system classifies new cases into the future. The Sherpa uses a static knowledge base, which does not change with time. We use an incremental belief updating procedure, which does not require all the data to be available once the conditional probabilities are calculated. Let \( \text{response}_n = \text{response}\#1, \text{response}\#2, \text{response}\#3 \) be set of \( n \) vectors of sequences of responses that were observed in the past, and \( \text{response} \) denote a new set of vectors each with 3 responses. The conditional probability can be updated as follows\(^{13}\):

\[
P(\text{Suspect} = i \mid \text{response}_n, \text{response}) = P(H \mid \text{response}_n) \frac{P(\text{response} \mid \text{response}_n, \text{Suspect} = i)}{P(\text{response} \mid \text{response}_n)}.\]

Assuming conditional independence, taking logarithms of non-constant variables and simplifying we have,
\[ P(\text{response}|\text{response}_n, \text{Suspect}=i) = P(\text{response}|\text{Suspect}=i), \text{ and} \]

\[ \log P(\text{Suspect}=i|\text{response}_n, \text{response}) = \log P(\text{Suspect}=i|\text{response}_n) + \log P(\text{response}|\text{Suspect}=i). \]

The above “log-likelihood” equation makes updating conditional probabilities extremely easy and BNS approach more suitable for dynamic learning.

6. The Distributed Systems Architecture For Implementation of BNS

The primary reason for web-enabling BNS is to reduce the communication between 30 analysts and 2 human agents. The current system leads to inefficiencies. By enabling the Sherpa over the internet, the communication requirements between 30 analyst and 2 human agents are lowered. The web-enabled BNS can allow the analyst analyze the data without contacting human agents, thereby increasing the efficiency of suspect identification.

There are two common approaches to web enabling the BNS system – enable the BNS on the server or enable it on the client\(^9\). When it is desired to enable the BNS on the client, a web interface can be wrapped around it. This requires running the BNS interface, and a Common Gateway Interface (CGI) program on the server to process the data from the HTML forms. One of the challenges related to the CGI programs is that CGI programs terminate after one action is performed. Since the interface agent may have multiple forms or pages for user to respond to, CGI program must be run multiple times\(^9\). Further, when there are many users running on the CGI server simultaneously it is imperative that the interface agent:

1. identify the individual user,
2. remember the user’s input,
3. save all the data,
4. send a new page to the user’s browser and

5. terminate.

The CGI terminate and restart process is the state of the art approach, which uses temporary files unique to each user to store the user specific data. A continuous (non terminating) CGI type processing is available, but it requires significant server resources to process each user step\textsuperscript{10}.

For a few users the BNS interface may be enabled on the server. The problem of enabling an BNS on server is that each user puts additional load on the server, and the server eventually slows down. Chang and Hsu\textsuperscript{5} propose an architecture for such a situation. The architecture proposed by Chang and Hsu\textsuperscript{5} is shown in Figure 9.

![Figure 9: Concept Learning Interface in Multi-User environment](image-url)
The interface agent in a multi-user environment can be located at the server. Users can access the standalone application server through their browsers and Common Gateway Interface (CGI). The historical data and the learned concepts will be stored on application server as well.

We propose a TCP/IP network and socket-based communication framework for implementing the proposed BNS. The advantage of using socket-based networking is that it can be used as a traditional distributed computing framework as well as a web based computing framework. Our framework, as illustrated in Figure 10, uses at least 4 processors. The first three processors are clients and the fourth processor is the server.

![Figure 10: A Socket Connection Based Distributed Computing Framework for BNS](image-url)
The agents can have their own operating systems, which may or may not be same. We propose that the agents use Java programming language when communicating with the server. Java programming language is operating system independent and is very flexible. The three agents can perform their local processing and come up with a recommendation, which will serve as an input for the Bayesian network. The recommendation can be sent to the server using the PrintWriter class in the Java programming language. Figure 11 lists the code that each agent must implement. The bold font indicates the items that are specific to the application. For example, IP address, will be the IP address for the server and port# is the agreed upon port (between clients and server) at which communication will take place.

```java
public class Agent {
    public static void main(String[] args) {
        try {
            Socket connectToServer = new Socket("IP address", port#);
            PrintWriter osToServer =
                new PrintWriter(connectToServer.getOutputStream(), true);
            while (true) {
                //Code for the agent classification
                osToServer.println(Classification);
            }
            catch (IOException ex) {
                System.err.println(ex);
            }
        }
    }
}
```

Figure 11: Java Code for the agents in Distributed Computing Architecture
The server, on the other hand, must implement concurrent processing by using multiple threads. Since the server will communicate with three clients, it will allocate separate memory and thread for handling each client’s input. The information from each client will be stored in separate thread memory by the server. The server will store this information by using BufferedReader class in Java programming language.

Figure 12 illustrates the code that the server should implement. The bold font indicates the items that are either specific to the application or should be added by the programmer.

The proposed distributed computing architecture provides flexibility and scalability that is currently lacking in the standalone system the Sherpa. Further, the proposed architecture allows for multiple operating systems, which will allow the users to continue to use the operating system they are comfortable in using. Thus, from implementation standpoint, managers are less likely to see organizational resistance.
public class BayesianNetworkServer
{
    static final int MAX_THREAD_LIMIT=3;
    public static void main(String[] args)
    {
        try
        {
            ServerSocket serverSocket = new ServerSocket(port#);
            int i = 0;
            ThreadGroup g = new ThreadGroup ("serving agents");
            while (g.activeCount() < MAX_THREAD_LIMIT)
            {
                Socket connectToClient = serverSocket.accept();
                Thread t = new Thread(g, newThreadHandle(connectToClient, i));
                t.start();
                i++;
            }
        }
        catch(IOException ex)
        {
            System.err.println(ex);
        }
    }
}

class ThreadHandle extends Thread
{
    private Socket connectToClient; // A connected socket
    private int counter; // Number the thread
    public ThreadHandle(Socket socket, int i)
    {
        connectToClient = socket;
        counter = i;
    }
    public void run()
    {
        try
        {
            BufferedReader isFromClient = new BufferedReader( new InputStreamReader(connectToClient.getInputStream()));
            while (true)
            {
                StringTokenizer s=new StringTokenizer(isFromClient.readLine());
                //Bayesian Belief Updating Code here
                System.out.println("Classification is " + Classification);
            }
        }
        catch(IOException ex)
        {
            System.err.println(ex);
        }
    }
}

Figure 12: Java Code for the Multi-Threaded Bayesian Network Server
7. Conclusions and Summary

The three main objectives of our research were: 1) to propose a hybrid BNS system that dynamically learns and adjusts to the changing environment, 2) to statistically compare the performance of the proposed system with the existing system and 3) propose a distributed systems architecture for proposed hybrid BNS. Our research objectives were achieved through the development of a hybrid BNS. The strategy for possible web-based implementation was described. Our results indicate that:

♦ DAI based hybrid BNS provides a better approach to knowledge management where information, resources or expertise are naturally distributed.

♦ Computerization is an important means of improving drug investigation. A web-enabled hybrid BNS developed in our research can strengthen drug intelligence (analysis and investigation).

♦ The knowledge management and data analysis in drug crime intelligence is the most crucial step. There is a need to design more efficient and effective information systems for drug intelligence.

The proposed distributed systems architecture provides several advantages. The advantages are scalability, heterogeneity, resource sharing and fault tolerance. Since Bayesian network can update beliefs when partial information is available, the system can continue to work while one agent is malfunctioning.

One of the issues, ignored in this research was that of security. The agents assume that the network communication is secured. When management believes that the communication is not secured, the communication between agents and server may be
required to be encrypted. Java provides several security manager and Java Cryptography Extension (JCE) classes to ensure secure transmission between client and server. The use of JCE will need extra coding in the proposed agent and Bayesian network server classes proposed in Figure 11 and Figure 12.

8. Acknowledgements

We would like to thank an anonymous reviewer for his/her invaluable comments.

References


Appendix: Probabilistic and Discriminant Models for Classification

Probabilistic Model

Duda and Hart\(^7\) provide mathematical foundation for the theory of classification. Using Bayesian theory, it can be argued that for a sample space of \(X \subseteq \mathbb{R}^n\) which is divided into \(k\) groups, the prior probabilities and conditional probability density function can be written as follows:

\[
P(\Omega_j) = \text{the prior probability that a randomly selected object belongs to group } j; \\
f(x|\Omega_j) = \text{conditional probability density function for } x \text{ when its group membership is known to be } \Omega_j.
\]

Using Bayes theorem, the posterior probability \(P(\Omega_j|x)\) can be written as follows:

\[
P(\Omega_j | x) = \frac{f(x, \Omega_j)}{f(x)} \\
= \frac{f(x|\Omega_j)P(\Omega_j)}{f(x|\Omega_1)P(\Omega_1) + f(x|\Omega_2)P(\Omega_2) + \ldots + f(x|\Omega_k)P(\Omega_k)} \quad \forall j = 1, 2, \ldots, k.
\]

Let a particular \(x\) is given and is to be assigned to a group. Further, let \(C_{ij}(x)\) be the cost of assigning \(x\) to group \(i\) when it actually belongs to group \(j\). Thus the expected misclassification cost of assigning in \(x\) to group \(i\) is formulated as\(^1\):

\[
Cost_i(x) = \sum_{j=1}^{k} C_{ij}(x)P(\Omega_j | x), \quad \forall i = 1, 2, \ldots, k.
\]

When it is desired to minimize misclassification costs, the decision-maker’s objective function can be represented as:

\[
\text{Minimize } \int_{x \in X} \text{Cost}(x)f(x)dx.
\]
It can be shown that the following Bayesian decision rule minimizes the total misclassification costs for a given vector \( x \),

\[
IF \ Cost_i(\ x) = \arg \min_{i=1,2,...,k} \ Cost_i(\ x) \ Then \ Group = \ \Omega_i.
\]

Considering zero misclassification cost for correct classifications and unit misclassification cost for true misclassification, the above rule can be simplified as,

\[
IF \ P_i(\ \Omega_i | \ x) = \arg \max_{i=1,2,...,k} P_i(\ \Omega_i | \ x) \ Then \ Group = \ \Omega_i.
\]

Using Bayes theorem, we can write

\[
\arg \max_{i=1,2,...,k} P_i(\ \Omega_i | \ x) = \arg \max_{i=1,2,...,k} \frac{P(\ x | \ \Omega_i)P(\ \Omega_i)}{P(\ x)}.
\]

Since \( P(\ x) \) is a constant, we drop it from the above equation, which leads to the following maximum a posteriori (MAP) hypothesis:

\[
\arg \max_{i=1,2,...,k} P_i(\ \Omega_i | \ x) = \arg \max_{i=1,2,...,k} P(\ x | \ \Omega_i)P(\ \Omega_i).
\]

Traditional classification systems that attempt to maximize total correct classifications use MAP hypothesis as their classification criterion.

**Discriminant Analysis**

The discriminant analysis procedure constructs a linear discriminant function by maximizing the ratio of between-groups and within-groups variances. For a binary classification problem, the discriminant function can be written as follows:

\[
D(X) = X'\sum^{-1}(\mu_1 - \mu_2) - \frac{1}{2}(\mu_1 - \mu_2)'\sum^{-1}(\mu_1 + \mu_2)
\]

where \( \mu_1, \mu_2, \) and \( \sum' \) are mean vectors for group 1, group 2 and inverse of common covariance matrix respectively.