Predicting Manufacturing Aptitude using Augmented Reality Work Instructions

Anastacia MacAllister, Eliot Winer, Jack Miller
Iowa State University
Ames, IA
anastac@iastate.edu, ewiner@iastate.edu, jackm@iastate.edu

ABSTRACT

The complexity of manufactured equipment for the U.S. military has increased substantially over the past decade. As more complex technology is integrated into battlefield equipment, it is more important than ever that workers manufacturing this equipment have the necessary skills. These specialized manufacturing skills require careful workforce selection and training. However, traditionally, workers are assigned roles based on instructor evaluation and qualitative self-assessments. Unfortunately, these assessments provide limited detail about a candidate's aptitude. By using more detailed data captured from assembly operations, a more complete profile of an operator's skills can be developed. This profile can then guide assignment of a worker to maximize productivity. This paper develops a Bayesian Network (BN) to predict worker performance using data captured from 75 participants via augmented reality guided assembly instructions. Information collected included step completion times, spatial abilities, and time spent on different assembly operations. For analysis, participant data was divided into training and testing sets. The data was mined for trends that could statistically predict measures of performance like errors or completion time. Based on these trends, the training set was used to construct the BN. The authors found that the model could predict some aspects of performance accurately, such as assembly completion time in the testing set. While these results were encouraging, further analysis demonstrated the network was biased by probabilities that were greatly influenced by the number of data points present in a category. The results highlight that, with small data sets, there is often not enough observed evidence to produce accurate predictions with BN. This suggests that a method of data simulation or generation is required to increase the number of training set samples. This would enable powerful BN tools to be used in real world manufacturing applications were collecting hundreds-of-thousands of data points is not feasible.

ABOUT THE AUTHORS

Anastacia MacAllister is a graduate student in Mechanical Engineering and Human-Computer Interaction at Iowa State University's Virtual Reality Applications Center. She is working on developing Augmented Reality work instructions for complex assembly and intelligent team tutoring systems. Anastacia was, also, one of the recipients of the 2016 Fred Lewis I/ITSEC scholarship.

Eliot Winer, Ph.D., is an associate director of the Virtual Reality Applications Center and professor of Mechanical Engineering and Electrical and Computer Engineering at Iowa State University. He is currently co-leading an effort to develop a next-generation mixed-reality virtual and constructive training environment for the U.S. Army. Dr. Winer has over 15 years of experience working in virtual reality and 3D computer graphics technologies on sponsored projects for the Department of Defense, Air Force Office of Scientific Research, Department of the Army, National Science Foundation, Department of Agriculture, Boeing, and John Deere.

Jack Miller is a graduate student in Mechanical Engineering and Human-Computer Interaction at Iowa State University's Virtual Reality Applications Center. He is working on developing Augmented Reality work instructions for complex assembly.

Predicting Manufacturing Aptitude using Augmented Reality Work Instructions

Anastacia MacAllister, Eliot Winer, Jack Miller
Iowa State University
Ames, IA
anastac@iastate.edu, ewiner@iastate.edu, jackm@iastate.edu

INTRODUCTION

With the increasing prevalence of commodity sensors and computing devices data can now be collected for relatively inconsequential costs and effort. Collecting data on a manufacturing process, customer satisfaction, or disaster response can now lead to a wealth of information for those dedicated enough to search for it. Through analyzing multivariate relationships between measured variables in collected data, organizations can use these relationships to help guide important decisions. These decisions are accomplished by utilizing machine learning techniques that can make accurate predictions and forecasts based on trends, even in novel situations. Not only can these machine learning tools aid in decision making, they can also be used to understand how different decisions or events impact a system as a whole. Companies like Amazon, Google, and Microsoft are using machine learning techniques to gain a competitive edge and help improve their product offerings (Biewald, 2016; Reese, 2016; Wilder, 2016). One such popular machine learning approach is Bayesian Networks (BN), based on powerful Bayesian statistical theory (Bayes, 1763). Bayesian Networks utilize the adaptive nature of Bayesian statistics to represent relationships between events in a compact, and easy to understand manner. While Bayesian statistics and networks are very powerful predictive tools, they often require hundreds-of-thousands or millions of data points to accurately model complex situations.

While using sensors to collect data is becoming more cost efficient, in real-world cases one cannot often collect enough data points to use machine learning techniques like BNs. In some domains, like manufacturing or battlefield training, events of interest may only happen a handful of times throughout the year. As a result, collecting thousands of unique data points is not possible. One specific example of the data volume issue is aircraft manufacturing. The process is very complex and involves many different collaborators from union labor to dozens of suppliers all impacting the finished product. Specifically, worker suitability can significantly impact assembly and manufacturing process outcomes. The ability to assign the correct worker to a job could provide a competitive edge by making sure their skills are suitably matched with a task (Ong, Ato, Umar, & Oshino, 2016). However, there might only be a handful of planes produced every month, meaning there is not enough worker data to construct a model to predict competencies (BBC, 2015). This means that a BN is not an ideal option because of the limited data available for training the network. The lack of data means there is not a way to accurately understand the relationships between variables, to predict how changes might impact the production process. As a result, these types of complex processes cannot make use of the powerful predictive analytics of BN because of the lack of data. The work in this paper begins exploring how small quantities of data can be used to construct a BN to accurately predict outcomes of a manufacturing operation. The data gathered was from an augmented reality guided assembly process that logged user performance and interaction with the assembly. This logged data was then analyzed for trends which were used to construct and train a BN. The goal of the network is to predict how well someone would do, in terms of errors and completion time, on the assembly. This work was made challenging by the small number of data points collected, 75, and the variation associated with human subject data. This paper presents the background of BN, the methods used to collect data, how the data was used to construct a BN, and the result of testing the network. In addition, ideas for improving the accuracy of the network while still using small amounts of data are discussed.

BACKGROUND

Statistical methods have long been used to help make sense of data and predict the likelihood of an event when provided with certain parameters. Statistical theory is used in areas spanning from reliability analysis to scheduling airline flights (Jacobs et al., 2012; Muller, 2003). The reason statistical methods are used increasingly, especially today, is their ability to suggest courses of action based on previously collected data. These suggestions benefit from

the ability to look at far more relationships between variables than humans can and provide decisions that are more unbiased (De Martino, Kumaran, Seymour, & Dolan, 2009; Hastie, 2001). Companies like Apple, Google, Amazon, and Netflix make use of statistical analysis every time they suggest a replacement word with auto correct, complete a search request, suggest a new product to order or show to watch. These real-time decisions are made possible by automated statistical analysis and machine learning algorithms. This type of customized real time decision making was unheard of twenty to thirty years ago and is made possible through using machine learning. Moving forward, leveraging this type of powerful analytics outside of the consumer realm is necessary in areas like the military to ensure the continued evolution of the warfighter.

Bayesian Statistics

Powerful machine learning techniques using Bayesian Networks are made possible due to the resurgence of Bayesian statistical methods. Bayes Theorem, shown in Equation 1, is unique from traditional statistical methods because it allows the incorporation of background information called the prior probability (Bayes, 1763). The term $P(A|B)$, called the posterior probability, represents the probability of A given that B occurs. The goal is to use what is known to calculate this unknown value. To calculate it involves using what is known on the right side of the equation. The term $P(B|A)$, reads the probability of B given A, is the likelihood of event B when A has already occurred in the population sample data. The term $P(A)$ is the prior probability of the event A happening anywhere in the population sample. The denominator term $P(B)$ is the probability of B occurring at any point in our population sample, which can be when A is observed or when A is not observed. Often in practice this term is dropped since the probability values are being compared and this term does not impact the results. For sake of space, this term is not explained. For a more in depth derivation see (Bayes, 1763).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

The inclusion of a prior differs from traditional statistical likelihood based approaches. These traditional approaches mainly estimate probabilities of events based on the observed sample population data (Orloff & Bloom, 2014). The introduction of the prior probability term allows for a correction or smoothing of the observed data described by the likelihood. Often this prior probability is thought of as an expert specified term. The combination of the prior and the likelihood give the posterior probability of an event occurring. Figure 1 shows an example of a possible likelihood distribution and a prior. Figure 1 shows the likelihood distribution, created from the data observed, with a strong probability of an event around x equals four. However, the prior distribution shows less certainty in the event happening at x equals four. The prior also has a wider variance than the likelihood distribution. The difference between distributions could be because the likelihood is over confident in the probability of an event occurring because the population data set that generated the distribution may not have included outlier data points. The prior distribution, created by an expert based on their experience, might consider that there may be outliers and thus decreases the probability of an event occurring at x equal to four. Using Bayes Theorem in Equation 1**,** the two distributions can be combined to provide a best estimate of the probability of an event occurring for a given x. Combining the likelihood and prior is a powerful strategy that allows for prediction of probabilities in areas where there is little data available, but experts know what the probabilities of certain events are.

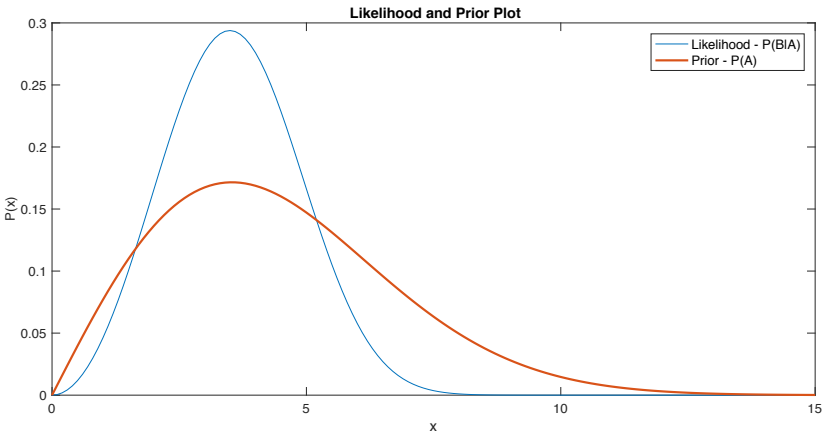

Figure 1. Likelihood and Prior Plot Example

Bayesian Networks

Conceptually, Bayes' Theorem, is straight forward when there are few events and few variables. In a simple problem, to calculate the probability of an event, multiply the prior by the likelihood. However, determining the outcome is challenging when there are multiple events with multiple

variables for each event. Representing the relationships between variables, becomes very complex, very fast. Helping to alleviate this problem, Bayesian Networks allow for the representation of dependencies and relationships between variables (Stephenson, 2000). This section describes the theory behind Bayesian Networks along with how to construct and solve for a network.

Let us start with an example. A quality control engineer is provided with a data set D that describes technicians' effectiveness at repairing armored vehicles. This data set is made up of n data points with each data point having $\mathbf{X} = \{v_1 \dots \dots v_i\}$ variables, see Equation 2. The variables in each data point represent data like vehicle type, technician experience, oil type used, repair time, use environment, etc. The engineer is tasked with coming up with a maintenance schedule assigning technicians to a vehicle, matching their skills with the type of maintenance. The engineer must determine how to model the relationships between the variables to properly assign a technician to each repair.

$$D = \begin{cases} \mathbf{X}_1 = \{v_1 \dots \dots v_i\} \\ \quad . \\ \quad . \\ \quad . \\ \mathbf{X}_n = \{v_1 \dots \dots v_i\} \end{cases} \tag{2}$$

Looking at the data, as is, the engineer must work with a joint distribution to understand how all the data behaves. In short, the engineer cannot separate out what variables might impact another without looking at all the variables. To understand all the relationships between the variables, the engineer must calculate 2^i different values, where i is the number of variables for each data point. This requires significant computing power for large multivariate data sets and it can be difficult for a human to interpret these causal relationships to make informed decisions.

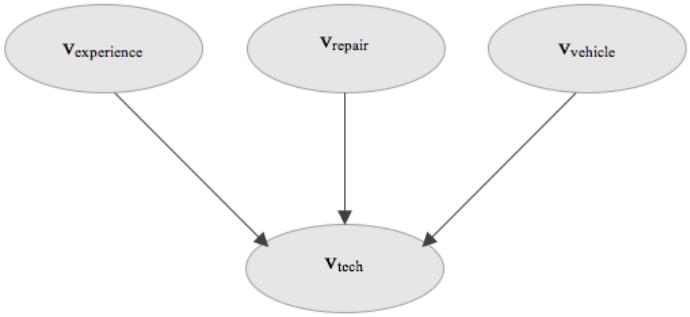

Figure 2. Example Bayesian Network for Technician Selection

If there was a way to visually represent the important causal relationships in the joint distribution, relationships between variables would be much easier to understand and manipulate. Fortunately, Bayesian Networks provide a way to do just that through Directed Acyclic Graphs (DAGs). These graphs are made up of vertices and edges. Vertices, also known as nodes, represent the variables that make up the data points and are represented as ovals in Figure 2. Edges denote the causal relationships between the vertices and are represented as directed arrows in Figure 2, where the vertex at the tail of the arrow is said to cause the vertex at the end of the arrow. These graphs allow for the direct visual representation of different variable dependencies, eliminating the need to interpret complex joint probability distributions. An example DAG for the technician selection problem is shown in Figure 2. This graph shows three causal variables, technician experience, type of vehicle repair and type of vehicle. These variables are known to be independent of one another. Meaning that the value of experience does not impact the value of the vehicle variable. However, these three variables cause or impact the selection of a technician.

Using the network topology displayed in Figure 2 the joint distribution can be rewritten as the product of individual probabilities**.** The general form of this equation shown in Equation 3. This equation denotes that instead of having to calculate the joint probability distribution to find out the probability of an event occurring, only the parents of a vertices need to be calculated, since only their values impact the result. Parent vertices, or nodes, are vertices at the start of a directed edge leading to a child, or dependent, node. In Figure 2, $v_{experience}$, v_{repair}, and $v_{vehicle}$ are parents of v_{tech}. A parent-child network representation reduces the number of required values. The computed values now required are no more than $i * 2^k$, where i is the number of vertices and k is the max number of parents of any vertex in the network.

$$P(v_{1,} \dots\dots, v_i) = \prod_{1}^{i} P(v_i | parents(v_i)) \tag{3}$$

Continuing with the technician selection example, the probability of an event occurring is given by Equation 4**.** In this equation, the probability of experience, repair, and vehicle are independent of other events because they have no parents. However, the probability of tech is dependent on its parents' experience, repair and vehicle. Using collected data, the probabilities of the events in Equation 4 can be calculated to provide a numerical probability of an event.

$$P(v_{experience}, v_{repair}, v_{vehicle}, v_{tech}) = P(v_{experience})P(v_{repair})P(v_{vehicle})P(v_{tech}|v_{experience}, v_{repair}, v_{vehicle}) \quad (4)$$

Before discussing how to solve for those event probabilities, it is important to mention that, in the example above, the network is already constructed. This, however, is not always the case. When creating a Bayesian Network to predict the probability of a given event there are four types of situations.

1. Known network topology and known relationships between vertices
2. Unknown network topology and unknown relationships between vertices
3. Known network topology and unknown relationships between vertices
4. Unknown network topology and known relationships between vertices

Learning network structure, or topology, from data is an active research area and requires significant amounts of data to ensure that the topology is accurate. Due to this paper dealing with small data sets, other methods are employed to construct the network topology. This paper focuses on calculating the relationships between vertices after the topology has been set (i.e. variation 3).

Continuing again with the technician example from above, let us consider a very small data set D consisting of four data points. The variables, or $v_i's$, measured for each data point are experience level (high or low), repair type (easy or difficult), vehicle type (common or rare), and technician experience (novice or expert). Each of these data points were actual repairs that were assigned based on an expert service manager's assessment of the situation. However, this service manager is retiring and the shop would like to capture their knowledge in an automated system. The data set gathered from the expert service manager's assignments is: $\mathbf{X}_1 = \{high, difficult, rare, expert\}$; $\mathbf{X}_2 = \{low, easy, rare, novice\}$; $\mathbf{X}_3 = \{low, easy, common, novice\}$; $\mathbf{X}_4 = \{low, easy, common, novice\}$. With these data points, the relationships between vertices in Figure 2 can be computed. To do that, a table of observed evidence must be created and grouped by prediction category, in this case the experience level of the technician assigned to the repair. In addition, the prior for each of the prediction categories must be computed. The prior calculation for discretized values is a simple probability calculation displayed in Equation 5, where n_c is the total number of observations that appear in a specific category (i.e. novice or expert) and n_n is the total number of obersvaitons in the data set D. Theoretically using this formulation, the prior probabilities for categories of a variable will sum to one. The prior probabilities for the vertices' who's value the network attempts to predict, in this case technician skill level, is shown in Table 1.

$$P(X) = \frac{n_c}{n_n} \quad (5)$$

The prior probabilities in Table 1 indicate that if we didn't know anything about the structure of the network or anything else about the data points, that when a job comes in it would be assigned to a novice 75% of the time. This occurs since 75% of the data in our training data set is falls into the novice category. However, looking at the data points there is more information available to help better assign a job to a technician based on additional variables. Using these variable values, the likelihood of selecting a technician category can be calculated given some information about the event.

Table 1. Technician Skill Level Prior Probabilities

Category	Prior Probability
Expert	$1/4 = .25$
Novice	$3/4 = .75$

To calculate the likelihood of an event given some evidence, the network must know what evidence it must base a decision on. The levels of evidence the network sees in each category are shown in Table 2. Note that the data described in Table 2 has been discretized rather than using continuous distributions like shown in Figure 1. This discretization is the process of taking continuous values and separating them into categories with an upper and lower bound. Discretizing values into categories makes the numerical calculations for solving a Bayesian Network less computationally intensive. There are numerous ways to discretize values, but one of the most popular is Hierarchical Clustering (Kerber, 1992).

Table 2 shows that using the training data set above, only one data point fell into the Expert technician category and there were three data points in the Novice category. The three data points in the Novice category exhibited two

different evidence states or unique combinations of the experience, repair, and vehicle variables. The goal, then, is to turn these observed evidence levels into probabilities that guide decisions about what category of technician a data point falls into based on the observed variables. To do this, the likelihood of an event given some evidence must be computed. Two traditional methods are called Laplace Smoothing and M-Estimate (Jiang, Wang, & Cai, 2007; MacKay, 1998; Williams, 1995). These methods are popular since they consider the probability of seeing a combination of evidence even if an event is not observed in the training data. This is helpful during the testing stage where a network may encounter novel data combinations.

Table 2. Laplace Smoothing Likelihood Calculations for Evidence

Technician Category	Experience	Repair	Vehicle	Count	Likelihood	Likelihood Non-Appearing
Expert	High	Difficult	Rare	1	$(1+1)/(1+2) = 0.67$	$\frac{0+1}{1+2} = 0.33$
Novice	Low	Easy	Rare	1	$(1+1)/(3+2) = 0.4$	$\frac{0+1}{3+2} = 0.2$
Novice	Low	Easy	Common	2	$(2+1)/(3+2) = 0.6$	

The Laplace Smoothing equation is shown in Equation 6. This equation calculates the likelihood, the term $P(B|A)$ in Equation 1, of evidence θ falling into class X and is read the probability of X given θ. The term n_c in this equation is the number of times a combination of evidence or variables appears in a category. For the expert category in Table 2, this value would be one since there is only one recorded occurrence of the unique combination of the $\mathbf{X}_1 = \{high, difficult, rare, expert\}$. The variable n is the total number of combinations in the category. In the example shown in Table 2, this value would be one since there is only one data point in the expert category. In the novice category, however, this value would be three since there are three data points falling under this classification. The term c is the number of categories. In the example above, this term would be two since there are two categories, expert and novice. The last term in the Laplace Smoothing equation is one. This is called the smoothing factor. Notice, in Table 2, for the Expert category there is less observed evidence than in the novice. As a result, since there is less known about this category, the Likelihood that testing evidence could appear in this category unknown to the trained model it is higher than the Novice category, which contains more observations. This probability is represented as a "Likelihood Non-Appearing" column in Table 2.

$$P(X|\theta) = \frac{n_c + 1}{n + c} \tag{6}$$

The M-Estimate method of calculating Likelihood is shown in Equation 7. This method has been shown, in some cases, to increase the accuracy of Bayesian Networks (Jiang et al., 2007). In this equation, the term n_c is the number of times a combination of evidence or variables appears in a category. The term k is the number of categories, p is the prior probability for a category, and n is the total number of combinations in the category.

$$P(X|\theta) = \frac{n_c + k * p}{n + k} \tag{7}$$

After Likelihood calculations, the network is considered trained. New points can be passed into the network for classification. For example, consider a new repair having the following measured states $\mathbf{X}_5 = \{low, hard, common, ?\}$. The shop wants to know who to assign this repair to so they put it though the trained Bayesian Network.

Table 3. Posterior Probability for Test Data Point

Category	Posterior Probability
$P(Expert\|low, hard, common)$	$P(Expert)P(low, hard, common\|Expert) = 0.25 * 0.33 \implies 0.0825$
$P(Novice\|low, hard, common)$	$P(Novice)P(low, hard, common\|Novice) = 0.75 * 0.2 \implies 0.15$

Table 3 contains the results of the Bayesian Network for the testing data point. The network predicts that the job should be given to a novice technician. Notice how much the prior probability impacts the result and how the network makes a prediction for a combination it has not encountered in the training data. Selecting an accurate prior is very important since it can greatly impact results. Selecting a prior probability is challenging with small amounts of data since little is known about the behavior of the data set numerically. Coming up with a way to have enough data in the

network to accurately account for behaviors a network might encounter, especially with users, is very challenging. This paper looks at how accurate a network is with a small amount of data. In addition, the work looks at how the distribution of the testing and training data falls onto the network.

Bayesian Networks in Manufacturing

Bayesian networks are powerful predictive tools. They are used extensively in industry and academia in areas from biological systems modeling to medical diagnosis (Aguilera, Fernández, Reche, & Rumí, 2010; Constantinou, Fenton, Marsh, & Radlinski, 2015; Molina, Bromley, Garcia-Arostegui, Sullivan, & Benavente, 2010). However, there is limited use of Bayesian networks in the manufacturing domain and even less work looking at human operators. This lack of human operator research is partially due to the challenges associated with collecting sufficient amounts of data to populate a network and because of the challenges associated with modeling human behavior. The limited work that exists detailing Bayesian Networks in manufacturing point to their ability to help accurately predict situations on the shop floor. Work by Jin, Liu, and Lin uses' optical sensor measurements to detect fixture faults in auto bodies and assess if the faults are outside tolerance (Jin, Liu, & Lin, 2012). Their work shows that BN can accurately predict when a part is out of tolerance. They conclude that BN are powerful and flexible tools for manufacturing, but that large amounts of data are required in some cases to produce accurate predictions. Work by Yang and Lee detailed a Bayesian Network for diagnosing faults in manufactured semiconductors (Yang & Lee, 2012). Their work uses automated sensors to measure characteristics of each wafer to predict if the wafer is good or bad. They found that by using a BN they could understand interactions between predictor variables. They, also, noted that the large amount of data and the training cycles for the network could be prohibitive to its widespread adoption. Mak, Afzulpurkar, et al. collected data form an automated soldering process using computer vision (Mak, Afzulpurkar, Dailey, & Saram, 2014). They used 330 samples to build a network that could predict when a faulty weld occurred. With their network, they could achieve 91% accuracy with a relatively small data set. However, while this data set was small for a traditional BN, the variation was very low. Using a small data set is easier with a low variation processes, this is not the case with human subjects. Dey and Stori used an even smaller data set to construct a BN (Dey & Stori, 2005). They collected 16 data points to predict variation in work piece hardness, stock size, and tool wear for a machining operation. Using their data, they achieved 80% accuracy. Although, again this work deals with consistent and repeatable machining data, which does not vary as much as human subject data.

Overall, the work presented above shows the potential of Bayesian Networks in the manufacturing domain. However, collecting enough data to build an accurate model is a challenge. This paper looks at using data collected from an augmented reality system to predict how well a participant can complete the assembly of a mock aircraft wing.

METHODS

Data Collection and Processing

Building the Bayesain Network first required data. Data was collected from an assembly task using Augmented Reality guided work instructions. This type of application was used to collect data because AR has shown considerable promise at helping improve assembly accuracy (Nakanishi, Ozeki, Akasaka, & Okada, 2007; Richardson et al., 2014; Wang, Ong, & Nee, 2016). In addition, the computer based AR system is an ideal platform for collecting the detailed process data required to predict manufacturing aptitude in a Bayesian Network. Participants in the study were asked to assemble a mock aircraft wing made of painted wood components and metal threaded fasteners. The study setup was designed to mimic a traditional work cell found in a manufacturing environment. To ensure that the assembly task aligned with operations found on an actual manufacturing floor, the instructions and assembly were created with input

Figure 3. Wing Assembly

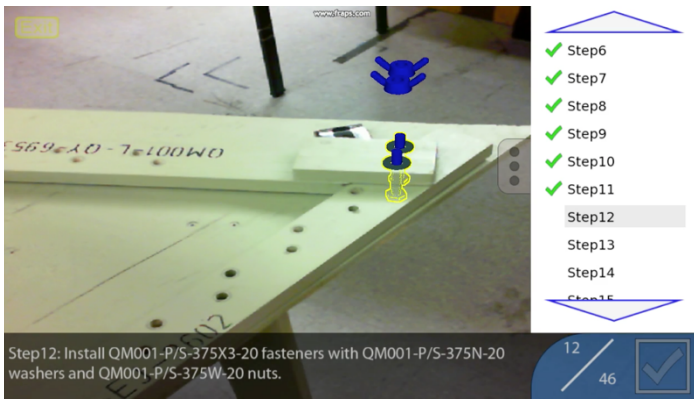

Figure 4. Augmented Reality Guided Work Instructions

from The Boeing Company. The wing was approximately 4 feet high and had a fixed base, shown in Figure 3. The wing had 12 large wooden components, three wires, and 14 fastener sets, which contained different varieties of bolts and nuts. For the study, AR instructions, shown in Figure 4, were delivered on a tablet computer fixed to a movable base. The AR application recoded when a participant moved between steps using a time stamped log file. These log files were then parsed to calculate how long participants spent on each step and how long they took to complete the assembly. Throughout the study observers sat behind a desk, where they recorded participant errors. For more information on this study, please see previously published work (Hoover et al., 2016; MacAllister, Gilbert, Holub, Winer, & Davies, 2016; Richardson et al., 2014).

Data parsed from the log files, that were used to train the network, consisted of time spent on different types of steps required to complete the wing (picking, placing, and assembly), paper folding score, errors committed, and total time. The goal was to use easily measured variables like step times and paper folding score to predict how quickly and how well an operator would complete the assembly. Ideally, being able to predict a participant's performance on a practice assembly, like the wing, could in the future be generalized to predict performance on a more substantial assembly. After this practice assembly, they could then be assigned to a job based on their projected abilities on a full assembly. If this were possible, a worker could come in to perform a short assembly that measured specific skills. After this practice assembly, they could then be assigned to a job based on their projected abilities on a full assembly. Current practices require workers to go through rigorous training taking weeks to months. Many workers drop out and only a fraction of those who compete it become proficient on the actual assembly line. This adds up to significant cost and time for the military or a company. BNs paired with detailed task data captured from AR work instructions offer the potential to significantly decrease these resources. Since worker suitability can significantly impact assembly and manufacturing process outcomes, the ability to assign the correct worker to a job could provide a competitive edge by making sure warfighters are assigned to tasks where they can make the highest impact (Ong, Ato, Umar, & Oshino, 2016).

Table 4. Linear Regression R^2 Values Between Variables

	Time	Fold Score	Picking	Placing	Assembly	Errors
Time		0.135	0.479	0.299	0.673	0.067
Fold Score	0.135		0.113	0.112	0.019	0.202
Picking	0.479	0.113		0.081	0.075	0.103
Placing	0.299	0.112	0.081		0.158	0.076
Assembly	0.673	0.019	0.075	0.158		0
Errors	0.067	0.202	0.103	0.076	0	

Constructing the Bayesian Network

After data were collected, before network training could occur, it had to be analyzed for trends. This analysis was necessary due to the small data set, meaning research methods that construct network structure could not be used. Data trends found through analysis helped establish the network structure representing the causal relationships between variables that impact a person's number of errors committed and total completion time. To understand the relationships between variables, linear regression models were created to show the strength of influence of variables upon each other. By finding the relationships between each variable, it can be established which variables can be used to predict time and errors. The R^2 values, the values that describes the strength of linear relationship between variables, are shown in Table 4. Generally, a higher R^2 value shows that there is a strong linear relationship between the variables. However, a low R^2 value does not mean the model is inherently bad. Since this data is from a human subjects' study, it is expected that there is a high amount of variance within the data. Therefore, the authors looked for strong trends between variables and relatively high R^2 values when building the network structure.

From looking at the values in Table 4 and trying to predict overall time it is apparent that both picking and assembly step times display strong relationships. This suggests that completion time is dependent on the outcome of picking and assembly times. This means an edge should be drawn from the vertices of picking time and assembly time to total time. This is represented in the network structure shown in Figure 5. The regression model used showed that for Time vs Assembly Time ($p < 0.0005$) and Time vs Picking Time ($p < 0.0005$) there was a statistically significant relationship.

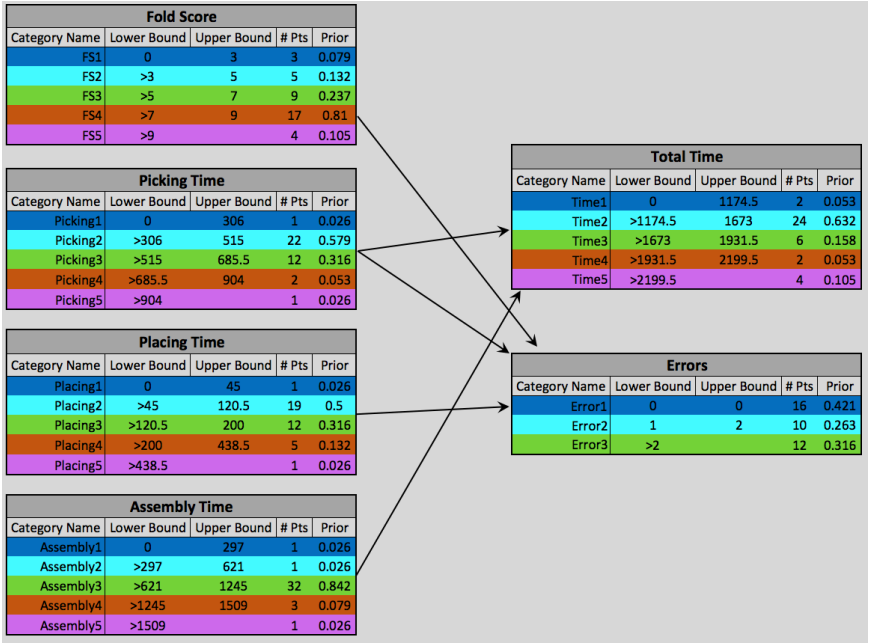

Figure 5. Bayesian Network Structure

When trying to predict the number of errors a participant would make, the linear regression models showed fold score, picking time and placing time as variables with the strongest relationships. Errors vs Fold Score ($p < 0.0005$), Errors vs Picking Time ($p = 0.005$) and Errors vs Placing time ($p = 0.017$) were all statistically significant and displayed the highest R^2 values. However, since there were relatively very few errors on average, there was high variance among the data. This could explain the low R^2 error correlation values, relative to time, and potentially cause issues modeling error behavior of participants in the system.

Table 5. Evidence Counts and Probabilities for Laplace Estimate

Predictor Category	Evidence-Assembly	Evidence-Picking	Evidence Count	Likelihood - Laplace Est	Likelihood - Non-Appearing Est
Time1	Assemb2	Picking1	1	0.285	0.142
Time1	Assemb3	Picking2	1	0.285	
Time2	Assemb3	Picking2	18	0.655	0.034
Time2	Assemb3	Picking3	5	0.206	
Time2	Assemb1	Picking5	1	0.068	
Time3	Assemb3	Picking3	4	0.454	0.090
Time3	Assemb3	Picking2	2	0.272	
Time4	Assemb3	Picking3	1	0.285	0.142
Time4	Assemb4	Picking3	1	0.285	
Time5	Assemb1	Picking3	1	0.222	0.111
Time5	Assemb4	Picking4	2	0.333	
Time5	Assemb5	Picking3	1	0.222	

Discretizing the Continuous Participant Data

To discretize the data, Hierarchical Clustering was used to group like values (Kerber, 1992). Grouping the data into clusters or categories makes the likelihood calculations less computationally intensive. In addition, discretized data works better with smaller data sets since there may not be enough data to construct a continuous probabilistic distribution. The results of the clustering algorithm are shown in network structure in Figure 5. Each vertex has multiple categories, instead of just high/low or expert/novice like in the background section example, with lower and upper bounds. Discretizing the data requires a participant's recorded numeric value for a variable to be assigned to a category where it fits inside the bounds. Each category houses multiple participant values that fall within its specific assigned range. As a result, instead of continuous values, categorical values are used for training networks. Figure 5

shows the upper and lower bounds for each category and how many of the 37 training points fell into each. The number of points in a category divided by the total number of total points was used to calculate the prior probability. Only half the data, which was selected randomly, was used to calculate these probabilities and train the network. The other half of the data was held out as a testing set.

While discretizing using, hierarchical clustering is preferred, for errors, existing standards dictated specific category bounds. The clustering algorithm did not follow this logic, so the bounds for the error vertex where set manually. To do this the participants were split into three groups based on their performance: high performers, moderate performers, and low performers. Values for these high, medium, and low performers was set based on expert evidence following manufacturing process specifications for Aerospace applications.

Training the Bayesian Network

Training the network required calculating the likelihood of observing specific evidence states within a predictor category. The predictor vertices in the network were time and errors. To calculate the likelihood of observing specific evidence, first, the number of times evidence combinations appeared were calculated for a predictor category as shown in Table 5 for time. From here, the number of observed combinations allowed calculating the likelihood using Equation 6 or Equation 7. Table 5 shows the observed evidence counts and the likelihood calculations using the training half of the data. This likelihood is multiplied by the prior probability during testing to produce the probability that a certain data point belongs to a certain time or error category. The non-appearing estimation column gives the probability that observed evidence is found in the category even if the specific combination does not appear in the training data. The same type of calculation was done for errors using combinations of fold score, picking time, and placing time to make up the evidence table. In addition, the network was trained using both the Laplace Estimate and M-Estimate likelihood estimator. The accuracy of the two methods is presented in the results section. Through comparing these two methods, insight can be gained if one of the metric is more accurate for small datasets.

RESULTS AND DISCUSSION

Table 6. Bayesian Network Accuracy by Likelihood Method

Likelihood Method	Accuracy – Time	Accuracy – Errors
Laplace Smoothing	.73	.38
M-Estimate	.70	.38

After using half of the data to train the network, the other half of the data was passed back through the network to gauge its accuracy at predicting a participant's completion time and errors. The simplest metric of gauging the accuracy of the two networks, Laplace and M-Estimate, is looking at how often a network categorized a testing data participant correctly. Table 6 shows the overall accuracy for each of the likelihood methods when attempting to predict a participant's time and errors from testing data. The table shows that Laplace Smoothing is slightly more accurate at predicting a participant's time category. Though, both methods were fairly accurate at predicting time, even with such a small data set. However, both methods struggled to accurately predict a participant's error category. For errors, each method was only slightly above the probability of randomly guessing between the three categories, which would be one-third.

Table 7. Laplace Smoothing Confusion Matrix for Time Prediction

	Predicted T1	Predicted T2	Predicted T3	Predicted T4	Predicted T5	Actual Count
Actual T1	0	1	0	0	0	1
Actual T2	0	26	0	0	0	26
Actual T3	0	6	0	0	0	6
Actual T4	0	2	0	0	0	2
Actual T5	0	1	0	0	1	2
Predicted Count	0	36	0	0	1	

While the overall accuracy can present a macro measure of performance, it does not tell the whole story. Looking at more detailed performance metrics can provide more insight into areas where the network encounters issues categorizing testing data points. One such method is called a confusion matrix. A confusion matrix shows the difference in the actual category and the predicted category. Table 7 shows the confusion matrix for the Laplace Smoothing likelihood calculation when predicting time. The table shows that of 37 training data points there were 36 data points predicted to fall into time category two. The table also shows that in the training dataset, only 26 of the

Table 8. Laplace Smoothing Confusion Matrix for Error Prediction

	Predicted E1	Predicted E2	Predicted E3	Actual Count
Actual E1	13	0	0	13
Actual E2	14	0	2	16
Actual E3	7	0	1	8
Predicted Count	34	0	3	

data points were actually in time category two, meaning the network over predicted participants would fall into time category two range of completion times. In fact, all six data points that were actually in time category three were incorrectly predicted to be in time two. The overall network accuracy for time was only high because just over 60 percent of the data points, after hierarchal discretization, fell into the time category two. Meaning that when the network assigned a time category two to a point, it had a greater probability of being correct. This bias towards assigning to time category two is due to the large prior probability associated with time category two, shown in Figure 5. This prior bias was due to the small sample sizes in the other categories. The time confusion matrix showed that the large prior probability biased the network to assigning time category two. The errors confusion matrix in Table 8 also shows that this is an issue for errors. Error category one has a slightly higher prior probability than the other error categories. Even through the difference is about ten percent, it seems to be enough to bias the network's assignment. In addition to the prior bias, the weak correlations found between evidence and predicted values for errors in Table 4 likely made predicting errors more challenging than time. To balance out the bias, either more data is needed to describe the evidence found in other categories or a different prior probability formulation is needed. A more balanced prior probability model would be less impacted by the imbalance in sample size. Moving forward, a distribution could be created from the data that better describes the prior probability for a given category. This could be accomplished by using fuzzy logic curves or by simulating data.

Table 9. Common Observed Evidence Between Testing and Training Data - Time

Predictor Category	Evidence-Assembly	Evidence-Picking	# Training	Likelihood Training	# Testing	Likelihood Testing
Time1	Assembly3	Picking2	1	0.285	1	0.333
Time2	Assembly3	Picking2	18	0.75	23	0.884
Time2	Assembly3	Picking3	5	0.208	3	0.115
Time3	Assembly3	Picking3	4	0.454	5	0.545
Time4	Assembly4	Picking3	1	0.285	1	0.285
Time5	Assembly4	Picking4	2	0.333	1	0.285

Another way to benchmark network performance, is to compare the distribution of the testing and training data within the predictor categories. By looking at how the data fell into the network categories, insight can be gained into how different data sets impact the likelihood of observing certain evidence. Table 9 shows the difference in Laplace Smoothing likelihoods for evidence levels in the training and testing data. Looking at the table, it is evident that the testing and training sets can have different likelihood values for the same levels of observed evidence. While these differences are slight. This can make a difference in accuracy since Bayesian Networks are a winner take all categorization tool. In some categorization decisions, the difference between two categories posterior distribution may be very small. This difference in the likelihood values between testing and training sets shows that there is a certain amount of variation in the small data set used. As a result, the network as is cannot fully describe the behavior of the data set. This can create issues when trying to accurately categorize or predict an assembler's skill level.

Table 10. Common Observed Evidence Between Testing and Training Data – Errors

Predictor Category	Evidence-Fold Score	Evidence-Picking	Evidence-Placing	# Training	Likelihood Training	Likelihood Testing	# Testing
Error1	FS5	Picking2	Placing2	2	0.157	0.25	3
Error1	FS4	Picking2	Placing2	4	0.263	0.125	1
Error1	FS3	Picking2	Placing3	1	0.105	0.25	3
Error1	FS3	Picking2	Placing2	2	0.157	0.187	2
Error1	FS3	Picking3	Placing4	1	0.105	0.125	1
Error2	FS4	Picking2	Placing2	3	0.307	0.263	4
Error3	FS2	Picking3	Placing3	1	0.133	0.181	1
Error3	FS3	Picking2	Placing3	1	0.133	0.181	1
Error3	FS3	Picking2	Placing2	1	0.133	0.181	1

The network described in this work is fairly simple, with only one layer of evidence and predictors. As tasks become more complex, the number of vertices and layers go up. As a result, the number of unique evidence combinations exponentially increases. With small datasets, seeing these combinations is challenging because there is not enough data in each of the evidence levels to statistically model. Above in Table 9, with two level of evidence predicting time, there was a large portion of the data sets that appeared in common categories. However, when the number of evidence levels is increased to three, in Table 10, only about half the data points fall into common evidence levels. This lack of common evidence between the testing and training data sets means that in many cases the non-appearing likelihood probability is being used to calculate the posterior, resulting in potentially lower categorization accuracies. This is a problem when working with human subject data, where, it is known that there is variation in the population sample. Points that do not fall into the common observed evidence in the testing and training sets are less likely to be categorized accurately. Those data points describe events or users that the system needs to be able to handle and predict skill levels for. However, with the current amount of data, there is not enough evidence to describe these values. As a result, the network will not accurately predict the skill of these users. To ensure that all users can benefit from such a system, more work needs to be done using the data available to simulate additional data. This additional data, can be used to train a network that is not biased by heavily weighted priors.

CONCLUSION AND FUTURE WORK

Bayesian Networks are a very powerful tool for modeling and predicting complex relationships between variables. These powerful tools have been shown to have applications in a wide variety of fields. However, the thousands or millions of data points required, mean that for some applications Bayesian Networks are not a viable option. This paper explores using Bayesian Networks to predict assembly accuracy and completion time. Data for the project was collected from an augmented reality guided assembly operation. The data from 75 participants was analyzed for trends to construct a Bayesian Network. Half of the participant data was used to train the network and the other half to test. Results indicated the network could predict assembly time with around seventy percent accuracy, but was only able to achieve thirty-eight percent error count accuracy. While these results were encouraging, further analysis demonstrated the network was biased by priors greatly influenced by the number of data points in a category. Further analysis, also, revealed that as network complexity increases the problems associated with small data sets increase. With small data sets, there are often not enough observed evidence combinations in categories to produce accurate predictions. The results suggest that for more complex problems, a method of data simulation or generation is required to increase the training set.

In the future, the authors would like to explore the ability to simulate data using the small collected data set as a seed. Using generated data grounded in user harvested data, powerful Bayesian Network tools can be deployed in non-traditional domains. Being able to use the powerful predictive tools of Bayesian Networks in areas like predictive maintenance or military training applications could help more accurately assign warfighters to tasks, improving success rates. In the end, the work presented above provides the first steps towards coming up with a strategy to begin using Bayesian Networks in manufacturing problems with small data sets.

REFERENCES

Aguilera, P. a, Fernández, a, Reche, F., & Rumí, R. (2010). Hybrid Bayesian network classifiers: Application to species distribution models. *Environmental Modelling & Software*, *25*(12), 1630–1639. http://doi.org/10.1016/j.envsoft.2010.04.016

Bayes, T. (1763). An Essay Towards Solving a Problem in the Doctrines of Chances. *Philosophical Transactions*, *53*(1764), 370–418. http://doi.org/10.1093/biomet/45.3-4.293

BBC. (2015). Boeing Delivers Record Number of Aircraft in 2015.

Biewald, L. (2016). How Real Businesses Are Using Machine learning.

Constantinou, A. C., Fenton, N., Marsh, W., & Radlinski, L. (2015). From complex questionnaire and interviewing data to intelligent Bayesian Network models for medical decision support. *Artificial Intelligence In Medicine*, *67*, 75–93. http://doi.org/10.1016/j.artmed.2016.01.002

De Martino, B., Kumaran, D., Seymour, B., & Dolan, R. J. (2009). Frames, Biases, and Rational Decision-Making in the Human Brain. *Science*, *313*(5787), 684–687. http://doi.org/10.1126/science.1128356.Frames

Dey, S., & Stori, J. A. Ã. (2005). A Bayesian network approach to root cause diagnosis of process variations, *45*, 75–

91. http://doi.org/10.1016/j.ijmachtools.2004.06.018

Hastie, R. (2001). Problems for judgment and decision making. *Annual Review of Psychology*, *52*(1), 653–683. http://doi.org/0066-4308/01/0201-0653

Hoover, M., MacAllister, A., Holub, J., Gilbert, S., Winer, E., & Davies, P. (2016). Assembly Training Using Commodity Physiological Sensors. *Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, (16159), 1–12.

Jacobs, T. L., Garrow, L. A., Lohatepanont, M., Koppelman, S., Coldren, G. M., & Purnomo, H. (2012). *Airline Planning and Schedule Development* (Vol. 169). http://doi.org/10.1007/978-1-4614-1608-1

Jiang, L. X., Wang, D. H., & Cai, Z. H. (2007). Scaling up the accuracy of Bayesian network classifiers by m-estimate. *Advanced Intelligent Computing Theories and Applications, Proceedings*, *4682*, 475–484\r1373.

Jin, S., Liu, Y., & Lin, Z. (2012). A Bayesian network approach for fixture fault diagnosis in launch of the assembly process. *International Journal of Production Research*, *50*(23), 6655–6666. http://doi.org/10.1080/00207543.2011.611543

Kerber, R. (1992). Chimerge: Discretization of numeric attributes. *Proceedings of the Tenth National Conference on Artificial Intelligence*, 123–128.

MacAllister, A., Gilbert, S., Holub, J., Winer, E., & Davies, P. (2016). Comparison of Navigation Methods in Augmented Reality Guided Assembly. *Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, (16075), 1–14.

MacKay, D. J. C. J. C. (1998). Choice of basis for Laplace approximation. *Machine Learning*, *33*(1), 77–86. http://doi.org/http://dx.doi.org/10.1023/A:1007558615313

Mak, C. W., Afzulpurkar, N. V., Dailey, M. N., & Saram, P. B. (2014). A bayesian approach to automated optical inspection for solder jet ball joint defects in the head gimbal assembly process. *IEEE Transactions on Automation Science and Engineering*, *11*(4), 1155–1162. http://doi.org/10.1109/TASE.2014.2305654

Molina, J. L., Bromley, J., Garcia-Arostegui, J. L., Sullivan, C., & Benavente, J. (2010). Integrated water resources management of overexploited hydrogeological systems using Object-Oriented Bayesian Networks. *Environmental Modelling & Software*, *25*(4), 383–397. http://doi.org/DOI 10.1016/j.envsoft.2009.10.007

Muller, C. (2003). *Reliability Analysis Of the 4.5 Roller Bearing*. Naval Postgraduate School.

Nakanishi, M., Ozeki, M., Akasaka, T., & Okada, Y. (2007). Human factor requirements for applying augmented reality to manuals in actual work situations. In *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* (pp. 2650–2655). http://doi.org/10.1109/ICSMC.2007.4413588

Ong, Y. D., Ato, S. S., Umar, V. K., & Oshino, K. H. (2016). Definition and Verification of Workers ' Aptitude Toward Assembly Tasks in Production Cells, *10*(1), 43–52.

Orloff, J., & Bloom, J. (2014). Comparison of frequentist and Bayesian inference. *MIT OpenCourseware*, 1–7.

Reese, H. (2016). Machine Learning: The Smart Person's Guide. Retrieved May 19, 2017, from http://www.techrepublic.com/article/machine-learning-the-smart-persons-guide/

Richardson, T., Gilbert, S., Holub, J., Thompson, F., MacAllister, A., Radkowski, R., … Terry, S. (2014). Fusing Self-Reported and Sensor Data from Mixed-Reality Training. In *I/ITSEC* (pp. 1–12).

Stephenson, T. (2000). An introduction to Bayesian network theory and usage. *Idiap Research Report*, 31. Retrieved from http://ftp.idiap.ch/pub/reports/2000/rr00-03.pdf

Wang, X., Ong, S. K., & Nee, A. Y. C. (2016). A comprehensive survey of augmented reality assembly research. *Advances in Manufacturing*, (July 2015). http://doi.org/10.1007/s40436-015-0131-4

Wilder, C. (2016). Rise Of The Machines Part 1: Google And Microsoft Stake Their Claims. Retrieved May 19, 2017, from https://www.forbes.com/sites/moorinsights/2016/04/15/rise-of-the-machines-part-1-google-and-microsoft-stake-their-claims/2/#139c66f717e2

Williams, P. M. (1995). Bayesian Regularization and Pruning Using a Laplace Prior. *Neural Computation*, *7*(1), 117–143. http://doi.org/10.1162/neco.1995.7.1.117

Yang, L., & Lee, J. (2012). Bayesian Belief Network-based approach for diagnostics and prognostics of semiconductor manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, *28*(1), 66–74. http://doi.org/10.1016/j.rcim.2011.06.007