Twin Copilot Using a Bayesian Network

Alora Greer
University of Alaska Anchorage
agreer5@alaska.edu

1. INTRODUCTION

This project was designed to create a solution to the problem of needing interactive guidance within a digital twin application, especially Kartorium. Kartorium is a local start-up company that provides easy-to-use digital twin platforms to users. Some users are companies such as an electric association as well as a cannery that need a digital representation of their facilities to access information. This is achieved by creating a project in Kartorium and building a scene to create the virtual representation. Once a project is created, users can choose between three scenes: Standard, Map, and Virtual Tour. Once in a scene, the users create features that go on the scene to create the desired virtual space for the desired environment. The problem is when someone is new to a digital twin application, they might need some guidance with building out their scene. This is where the Twin Copilot could help. Artificial intelligence assisting users could be more beneficial than a human. It would take away time from the Kartorium developers to help build scenes when the Twin Copilot could do most of the heavy lifting by making predictions based on past user input or generated fake user input data.

2. PROBLEM

Trying to navigate the Kartorium digital twin web application when I had no experience at all with digital twins or any related application was difficult. Having an agent make suggestions within the application would have saved me some time. Talking with the company, I learned that one of their goals is to create an onboarding feature so new users or users without experience can navigate this application. The more user-friendly this application can be, the more it can be used within companies with projects using Kartorium.

3. EXISTING SOLUTIONS

Finding existing solutions for an AI assistant for a digital twin application was difficult. However, there were some solutions for AI assistants using Neural Networks. The Authors of [1] created an Intelligent Personal Assistant (IPA) that predicts users' next actions and improves over time. They used neural networks to learn user behavior and make predictions about future actions. The data collection was users' interactions over time. The learning accuracy was 93.85% on 15 new users.

Another approach to predicting future offers in a negotiation setting was neural networks. This isn't directly related to user input on a web application. The authors wanted to predict the expected counteroffer according to past information. The model in this study had high performance. [2]

This study used Bayesian Networks to solve a problem for tutoring students. It is an intelligent agent that can adapt to student needs and learning characteristics over time. The Bayesian Network is used to estimate student knowledge and updates with more data. It can predict students' next actions. It demonstrated high performance and accuracy in predictive performance abilities. [3]

4. PROPOSED SOLUTION

The proposed solution was to create a heuristic-based model that uses basic decision rules. It could be a baseline basic model for new users since there would be no data for a new user for a Bayesian Network to be trained on.

Once there is more user data, there will be a transition to the Bayesian Network. That user's data would be trained on the Bayesian Network to make predictions for the following steps to help guide the user within the application based on past preferences. All according to a simple Bayesian Network design to ensure higher accuracy.

Now that the user is proficient, they will have more data, and the Bayesian Network will become more complex with more actions added. This is when the transition to a machine learning model (MLM) will occur.

After further research and implementation, the plan to move to an MLM was changed to a Neural Network instead. The heuristic-based model was dropped and switched to using a Bayesian Network trained on generated fake user input data until actual data was collected.

5. ADVANTAGES AND DISADVANTAGES

There are advantages and disadvantages to using a Bayesian Network versus a Neural Network. Based on my research, neural Networks are prevalent now and seem to be the first choice. That doesn't mean Bayesian Networks are not useful.

5.1 Advantages

5.1.1 Pros of Bayesian Networks

When using a Bayesian Network, it can be accurate without a large amount of data. Kartorium is a startup company, and no user data is collected at this point. One solution would be to wait while more data is collected or generate fake data.

It is less computationally expensive to run and that can be very beneficial for a startup. It is also more convenient to and more accessible to adapt.

The approach is transparent, and we know exactly how the model is going to work. There are no questions left unanswered.

5.1.2 Pros of Neural Networks

When considering Neural Networks, there are some pros and cons. The accuracy is higher when using a Neural Network on a more complex problem. The more complex users' actions can be, the harder it is for the Bayesian Network to predict. Using Neural Networks is beneficial for more complexity and could provide users with more personalized, more accurate preferences. If there is a large amount of user data, the Neural Network can handle it all and be accurate.

5.2 Disadvantages

5.2.1 Cons of Bayesian Networks

The accuracy of the predictions can be lower with more complexity. The network would need to stay somewhat simple to ensure accuracy. If the user wanted more predictions for specific actions outside of the network, they would be left unsatisfied. It may be less personalized.

5.2.2 Cons of Neural Networks

The computational expense is more and might not be ideal for a startup. It is more complex to use for simple problems and requires large amounts of data. Collecting large amounts of user data can take time and may not be necessary for the problem. It is less transparent and a black box, giving no insight into how it works.

6. METHODOLOGY

6.1 Implementation of Bayesian Network

Designing the Bayesian Network for Kartorium, I first started off by choosing simple user actions. When in a project, the user can build a scene. There can be multiple scenes in a project and there are three scene types: Standard, Map, and Virtual Tour. Once the user is in a Scene, they can upload a feature. There are many feature types, but I chose just the most common ones: Text, Image, Model, and GeoJSON. All these features except GeoJSON can be uploaded to all scene types. Therefore, the scene type influences the feature type. Lastly, when the user uploads a model, they can go into four different modes. They can move and resize the model with translate, rotate, and scale modes. The user can also measure the model. Refer to Figure 1. After creating the design I implemented it by using pgmpy.models Bayesian Network library.

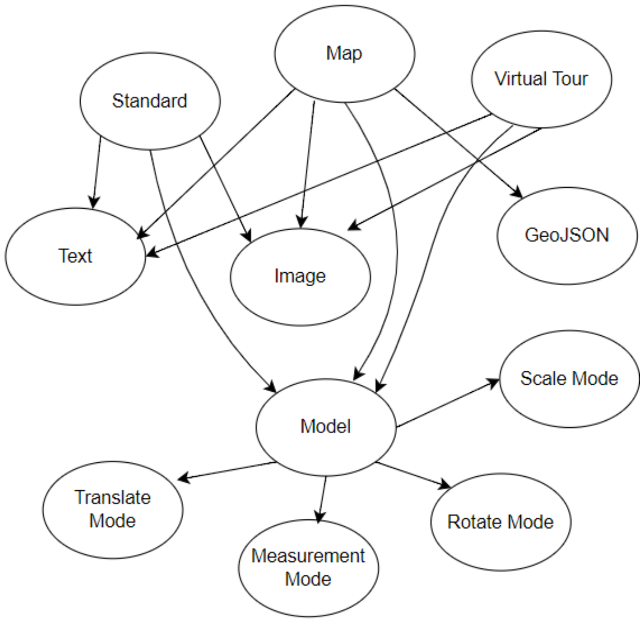

Figure 1. Bayesian Network for Kartorium Project creation. Scene Types are the parents of Feature Types, and Model Feature has four modes.

6.2 Dataset

I generated a fake user input dataset to create enough data to train the Bayesian Network. I started off by randomizing the user actions for each situation. I noticed that there wasn't much variation when doing so. I then decided to change the data in a way that made more sense in a real-world situation and how the user would use the Kartorium application. First, I generated fake data so fake users would choose Standard Scene 70% of the time. Then, when a user chooses a Map Scene, they pick GeoJSON 60% of the time. When they chose a Standard Scene, they picked a model to upload 60% of the time. Lastly, when they chose to upload a model they chose to measure it 80% of the time. Originally, I started off by just changing the amount of time they pick a GeoJSON. I realized it would be more interesting to make more changes. The example dataset table is seen in Table 1.

User ID	Scene Type	Feature Type	Model Feature
9020	Map	GeoJSON	Measure
3031	Map	GeoJSON	Measure
4449	Standard	Text	Translate
8843	Standard	Model	Translate
2980	Map	GeoJSON	Measure

Table 1. Table of dataset.

The fake data was generated for 10,000 users and 1,000,000 choices. Then the data needed to be binary, so it was one-hot encoded using pandas concat. Before training the model the data was spilt 80% for training and 20% for testing.

7. EVALUATION STRATEGY

Using the method in equation (1) for constructing the network. I then calculated the probability distribution.

$$p(x_1, \dots, x_n) = \prod_{i=1}^{n} p(x_i | x_{i-1}, \dots, x_1) \quad (1)$$

The Network was trained using the Maximum Likelihood Estimation method to learn the conditional probability distribution (CPDs) from the training data. The probabilities were then inspected and compared to test data using the Variable Elimination for inference. To calculate the accuracy of network predictions, the test data was compared to the real predictions of the test data. Specifically, the probabilities for GeoJSON given Map, Model given Standard and Measure given Model.

8. RESULTS

First, the CPDs were calculated for the parent nodes. When the original dataset was generated randomly, the three parent nodes had an equal probability of being chosen with a 1/3 probability. After I added the 70% preference to Standard, the probabilities changed. Standard had a 80% probability of being chosen, Map had 10%, and the virtual tour had 10%. The probability GeoJSON was picked given Map scene was 70%. The probability Model was picked given Standard scene was 69%. Lastly, the probability Measure was chosen given Model was 85%.

The accuracy metrics were computed by comparing the Bayesian Network's prediction against actual states in the test data, specifically, with probabilities for GeoJSON given Map, Model given Standard, Measure given Model, and Standard given Measurement. There was a 90% accuracy rate for GeoJSON given Map. A 77% accuracy rate for Model given Standard. A 85% accuracy rate for Measure given Model. Lastly, a 75% accuracy rate for Measure given Standard.

9. CONCLUSION

There is room for improvement in some regions of the network. The test data results can help find areas to improve, such as the measure given standard—the more complex the network, the less accurate the predictions. Given the fact that Kartorium doesn't have ready user input data available now, adopting this approach and using the Bayesian Network could be more beneficial. It solves the fundamental problem of having new users need some guidance with helping navigation of the application. Improving this model could help lead to more personalization and intuitive user experience. Also, implementing a Neural Network may be a future solution since it could be more accurate and more personalized.

10. REFERENCES

[1] Czibula, G., Guran, A. M., Czibula, I. G., & Cojocar, G. S. (2009, August). IPA-An intelligent personal assistant agent for task performance support. In *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing* (pp. 31-34). IEEE.

[2] Carbonneau, R., Kersten, G. E., & Vahidov, R. (2008). Predicting opponent's moves in electronic negotiations using neural networks. *Expert Systems with Applications*, *34*(2), 1266-1273.

[3] Wu, L. (2020). Student model construction of intelligent teaching system based on Bayesian network. *Personal and Ubiquitous Computing*, *24*(3), 419-428.