Project Report Traffic Flow Prediction Using Bayesian Models

Tanay Patel, Rajat Kumar Thakur, Abhi Tundya, Tejas Pakhale

Abstract—**This lab report highlights the use of probabilistic models to predict traffic flow as a solution to reduce congestion and its negative impacts, such as increased travel time and fuel consumption. By developing a data-driven system that forecasts traffic patterns, the project demonstrates how accurate predictions can improve traffic management, optimize routes, and alleviate congestion. The system's predictive capabilities offer practical such as reducing costs, and minimizing environmental impact.**

I. Introduction

Bayesian Networks are highly effective for traffic flow prediction due to their ability to model complex dependencies among variables such as traffic volume, road conditions, and temporal factors. These graphical models represent probabilistic relationships, allowing for the capture of conditional probabilities that are critical for understanding traffic dynamics. By employing Bayesian inference, the model can update predictions in real time as new data is introduced, facilitating accurate forecasts despite incomplete information. This capability enables proactive traffic management strategies, optimizing route decisions and alleviating congestion in urban environments.

II. Understanding

A. Bayesian Models

Bayesian models provide a framework for predicting complex systems through probability. These models utilize Bayes' theorem to update the probability of a hypothesis as new evidence is obtained. In traffic flow prediction, Bayesian Networks serve as a graphical representation of the relationships between factors influencing traffic dynamics.

Each node in the network represents a random variable, such as traffic volume, time of day, and weather conditions, while the directed edges indicate conditional dependencies. This structure enables the incorporation of prior knowledge and data, facilitating the estimation of conditional probability tables (CPTs) that define the relationships between variables.

Bayesian models effectively handle uncertainty, making them valuable in scenarios with incomplete or noisy data. By utilizing these models, practitioners can compute posterior probabilities of different traffic states, enabling data-driven decision-making. This capacity for real-time updating positions Bayesian models as a tool for traffic management.

B. Incremental Association Markov Blanket Algorithm

IAMB (Incremental Association Markov Blanket Algorithm) is an algorithm used for structure learning in Bayesian Networks. It focuses on identifying the Markov blanket of a target variable, which includes its parents, children, and other variables that make the target conditionally independent of the rest of the network. The IAMB algorithm incrementally refines the network structure by adding and removing edges based on statistical tests, ensuring that only relevant dependencies are maintained. This method enables efficient learning of the relationships between variables, facilitating the construction of a more accurate Bayesian Network that reflects the underlying data structure. Learners will gain insights into how to define the structure of Bayesian Networks, specify Conditional Probability Tables (CPTs), and perform inference tasks such as querying probabilistic queries and learning from data.

C. Hill Climbing Algorithm

Bayesian Networks often rely on domain knowledge or expert input to specify variable dependencies. However, they can also be learned directly from data. Learning algorithms such as structure learning and parameter learning enable the extraction of dependencies and CPTs from observational data.

Participants will understand the process of learning Bayesian Network structures and parameters from data, including algorithms such as constraint-based methods, score-based methods, and hybrid approaches.

III. Problem Statement

Traffic Flow Prediction: Urban traffic congestion increases travel times and fuel consumption. Current traffic management approaches are often inadequate for handling complex, dynamic traffic patterns. This project aims to develop a proactive, data-driven traffic management system using probabilistic models to predict traffic flows and congestion patterns

label=0.

1) **Data Collection and Preprocessing:**
   - Historical traffic data from National Databank Wegverkeersgegevens (NDW)
   - Street network data from Open Street Maps (OSM)
   - Study areas: Road networks of Amsterdam and Rotterdam
   - Aggregation of data for analysis

2) **Bayesian Network Construction:**

- Use of IAMB and HC algorithms for structure learning
- Parameters: Intensity, Travel Time, Traffic Speed, Vehicle Categories, Weekend/Weekday, Day of Week, Time of Day
- Model suitability evaluation using BIC, AIC, BDE, and k2 scores
- K-fold cross-validation for model validation

3) **Probabilistic Analysis:**

- Calculation of prior probabilities for each node
- Analysis of posterior probabilities to understand influences on traffic intensity
- Identification of major causes for increased traffic intensity using odds ratio

4) **Traffic Flow Prediction:**

- Short-term predictions for 30-minute and 60-minute intervals
- Use of continuous data for parameter learning (Hill Climbing algorithm)
- Forecast accuracy assessment using RMSE, MAE, and MAPE

5) **Street Network Analysis:**

- Analysis of road network characteristics for Amsterdam and Rotterdam
- Calculation of network statistics (nodes, edges, degree, intersections, etc.)
- Visual comparison of street network structures

6) **Model Evaluation and Comparison:**

- Comparison of IAMB and Hill Climbing algorithms
- Assessment of model performance for different time intervals and cities

7) **Conclusions and Future Work:**

- Evaluation of the Bayesian Network approach for traffic congestion modeling
- Recommendations for model improvement
- Suggestions for incorporating additional data sources
- Proposal for developing dynamic, spatial models

IV. Implementation

A. Hill Climb Algorithm

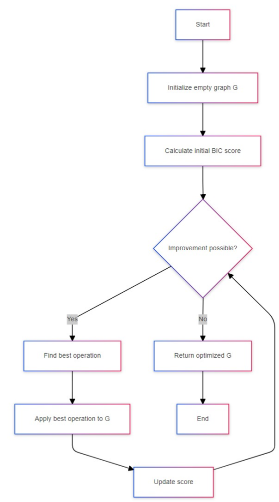

Figure 1: Caption

The Hill Climbing Algorithm is used to learn the structure of a Bayesian Network from the data. It improves the data by manipulating the edges between the variables in the network.

Algorithm 1 Hill-Climbing Algorithm for Bayesian Network Structure Learning

Input: Data D, Scoring function S, Set of possible structures $\mathcal{G}$
Output: Directed Acyclic Graph (DAG) G
Initialization:
  Start with an empty graph G_0 or a random graph G_0
$G \leftarrow G_0$
repeat
  Generate neighbors of the current graph by:
    1. Adding an arc between two variables
    2. Deleting an arc between two variables
    3. Reversing the direction of an arc
  Select the neighbor G_{new} that maximizes the score $S(G_{\text{new}}, D)$
  if $S(G_{\text{new}}, D) > S(G, D)$ **then**
    $G \leftarrow G_{\text{new}}$
  end if
until No improvement in the score is possible
Return G

B. Bayesian Prediction Model

Method: Bayesian Inference

Input: Prior probability $P(H)$, Likelihood $P(E|H)$, and Evidence E

Output: Posterior probability $P(H|E)$

1) **Prior:** Specify the prior probability $P(H)$, which represents the initial belief about the hypothesis H before observing the evidence E.

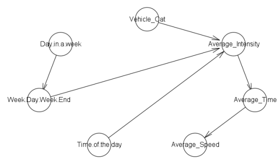

Figure 2: Graph of Bayesian network.

2) **Likelihood:** Compute the likelihood $P(E|H)$, which represents how likely the evidence E is, given the hypothesis H.
3) **Evidence (Marginal Likelihood):** Calculate the marginal likelihood or evidence $P(E)$ by summing over all possible hypotheses:

$$P(E) = \sum_{H'} P(E|H')P(H')$$

4) **Posterior:** Apply Bayes' Theorem to compute the posterior probability of the hypothesis H given the evidence E:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

This posterior probability represents the updated belief in the hypothesis H after observing the evidence E.

V. CONCLUSION

In this lab, we explored the concepts of graphical models, Bayesian inference, and classification in the context of educational data analysis. By constructing Bayesian Networks, we were able to model dependencies between student grades across courses and learn Conditional Probability Tables (CPTs) from data. Furthermore, implementing naive Bayes classification allowed us to predict internship qualification status based on student performance, initially assuming independence between course grades.

Through experimentation and evaluation, we observed the impact of considering dependencies between course grades on the performance of the naive Bayes classifier. While the classifier performed reasonably well without considering dependencies, incorporating dependencies led to improved accuracy and better alignment with real-world scenarios.

Overall, this lab provided valuable insights into the application of graphical models and probabilistic reasoning techniques in educational data analysis, highlighting the importance of considering dependencies for more accurate predictions and decision-making.