MTU, Department of Computer Science

COMP9016 Assignment #2

Dr Ruairi O'Reilly

October 30, 2024

1 Knowledge Representation

The purpose of this assignment is to assess the following **Learning Outcomes**:

- LO1 Appraise domain specific formalisms used in knowledge representation schemes.
- LO2 Compare and contrast current knowledge representation approaches integrated in systems relevant to AI.
- LO3 Select, apply and evaluate a knowledge representation scheme for a specified domain.
- LO4 Design and implement KR formalisms for a real-world data set.
- LO5 Interpret, critique and communicate the suitability of data visualisation techniques used in conjunction with the design of KR formalisms and the analysis of the resulting output.

To that end, you are going to build upon the theory discussed in lectures and the practical work carried out as part the course.

1.1 Logical Reasoning

Problem Statement: Define the following family relationships and property inheritance rules in first-order logic. Use these definitions to create a knowledge base and infer new relationships and properties.

1. **Ancestor Relationship**: A person is an ancestor of another if they are a parent or if they are a parent of one of the other person's ancestors.

2. **Cousin Relationship**: Two people are cousins if their parents are siblings and they are not the same person.
3. **Inheritance of Property**: Define a property, such as "has_blue_eyes," that can be inherited from a parent. Assume that if one parent has this property, there is a 50% chance their child inherits it.

1.1.1 Facts

Given the following facts:

- **Alice** and **Bob** are parents of **Carol**.
- **Alice** and **Bob** are parents of **Dave**.
- **Eve** is the spouse of **Dave**.
- **Carol** is the parent of **Frank**.
- **Carol** has blue eyes.

1.1.2 Tasks

1. **Define Rules and Facts in First-Order Logic**: Formulate rules for each relationship and property inheritance using first-order logic.
2. **Code the Knowledge Base**: Using `FolKB` from the `logic.py` file in the AIMA Python library, encode these facts and rules.
3. **Inference**:
   - Using forward and backward chaining, derive if **Frank** has blue eyes.
   - Derive if **Frank** has an ancestor with blue eyes.
   - Derive if **Carol** and **Eve** have a cousin relationship.
   - Derive all possible ancestor relationships in the knowledge base.
4. **Compare Inference Approaches**:
   - Compare the outcomes from forward chaining and backward chaining and describe any differences in efficiency or results.

1.1.3 Additional Instructions

- **Forward Chaining and Backward Chaining**: Use the AIMA library's implementations to apply forward and backward chaining and derive all possible inferences from the given facts and rules.

- **Implementation Hints**: Consider creating functions for each relationship type and property to encapsulate inference rules. For example, you may use predicates like `Parent(x, y)` or `has_blue_eyes(x)` to capture inheritance.
- **Reflect**: Briefly discuss how property inheritance can be represented in logical reasoning and the limitations of using logic-based approaches for probabilistic inheritance.

1.2 Bayesian Networks

See the set of Random Variables representing factors in an environmental and technological impact assessment:

- CarbonEmissions
- TechInnovation
- EcologicalFootprint
- CleanEnergyAdoption
- Urbanisation
- JobMarket

Your task is to develop a Bayesian model that encapsulates the relationships between these factors. Your model should be based on your interpretation of how these variables affect one another, and you should provide justification for the structure of your model. The above factors are **mandatory**, feel free to add an additional 2 to 4 factors if required.

Develop a Bayesian Network that models the relationships between these factors based on your understanding of how they interact. The structure of the network should be accompanied by a rationale that explains causal relationships, including causal chains, common causes, and common effects. Discuss these connections in terms of d-separation, describing how certain nodes become conditionally independent given others, and explain why that reflects real-world causal mechanisms.

1.2.1 For your Bayesian Network:

- **Graphical Representation**: Construct and visually present the Bayesian Network, illustrating the directional dependencies among variables. Ensure that the network structure reflects causal chains, common causes, and common effects where appropriate. For example, consider if *CarbonEmissions* and *Urbanisation* are both influenced by *TechInnovation* (a common cause), or if *CleanEnergyAdoption* acts as a common effect of both *TechInnovation* and *CarbonEmissions*.
- **Conditional Probability Tables (CPTs)**: Define the Conditional Probability Tables (CPTs) associated with each node in the network. The CPTs should capture probabilistic dependencies between variables as informed by the causal structure.

- **Querying the Network**: Demonstrate querying the network to retrieve specific probability values or to predict outcomes based on given evidence. Use d-separation principles to explain how the network isolates certain variable dependencies under given conditions.

1.2.2 Bayesian Networks: Utility, Pros and Cons

Write a detailed and concise description of Bayesian Networks, focusing on their utility in modelling complex probabilistic dependencies and causal reasoning. Describe the advantages of Bayesian Networks, such as their ability to represent conditional independence and perform efficient inference through d-separation. Discuss limitations as well, such as challenges in specifying accurate conditional probabilities in complex models. Where possible, support your explanation with relevant examples related to environmental or technological impact.

1.3 Implementation and Analysis of Naive Bayes Classifiers

In this task, you will be implementing Naive Bayes classifiers, which are probabilistic models used for classification tasks. The classifiers are based on the principle of feature independence given the class label, according to Bayes' theorem.

1.3.1 DATA SELECTION AND PREPROCESSING

Select two appropriate datasets with a manageable number of features and classes from the UCI Machine Learning Repository: `https://archive.ics.uci.edu/ml/index.php`. Extract a relevant subset from each dataset and perform the following analyses:

- Calculate the Prior probabilities for the classes in each dataset.
- Estimate the probability of the evidence within the dataset.
- Determine the likelihood of the evidence (the numerator of Bayes' formula).

Explain how each of these concepts relates to Bayesian networks and Naive Bayes classifiers. Discuss the suitability of the chosen datasets for Naive Bayes classification, and support your argument with appropriate visualizations.
Provide a detailed description of the datasets selected and their compatibility with Naive Bayes classifiers and probabilistic models in general.

1.3.2 NAIVE BAYES CLASSIFICATION

Following your review of a "learning.ipynb" Jupyter notebook, implement the Naive Bayes algorithm. Evaluate its performance on your selected datasets. Choose an appropriate version of the Naive Bayes classifier, if necessary, to suit the characteristics of your data. Illustrate the classifier's performance with relevant visualizations.

Conclude with a comprehensive explanation of the Naive Bayes Classifier, covering the key concepts of probability, conditional independence, and Bayes' theorem, as they apply to the algorithm. Your description should be clear, concise, and demonstrate a thorough understanding of the classifier's theoretical foundations.

2 Tips

- **Review** "logic.ipynb", "probability.ipynb", "learning.ipynb" and 'learning_apps.ipynb".
- **Tip for all Parts**: Do not provide masses of text lifted from the text/online or wikipedia. Be your own editor, focus on what is important and articulate that in a rationale way that is technically accurate. Demonstrating a capacity to explain complex topics in a clear and concise manner is indicative of a higher level of understanding and will be rewarded with an appropriate grade. Use results derived from experimental runs to back your claims and make sure your narrative is coherent and consistent.

3 Submission

You submission will contain:

- A 1800 word report ≤ 5 pages in ".pdf" format. Note: include name:student number in footer of document.
- A **single** ".py' file

Prepare your submission by archiving it as a single ".zip" file using the following naming convention.

"A2_COMP9016_<Surname>_<First name>_<Student Number>.zip"
e.g.
"A2_COMP9016_OReilly_Ruairi_R123456.zip"

3.1 Word count - 1800 word max

There are three distinct questions as part of this assignment Q1.1-Q1.3. In answering these it is important that you demonstrate an understanding of the underlying theory as applied in the context of your environment. There is a suggested word limit of 600 words per question, your ability to articulate your answer succinctly is indicative of a deeper understanding in itself. I **highly recommend** the use of well formatted tables for any experimental results and/or diagrams/visualisations if they aid the message you are trying to convey. Boilerplate text should be kept succinct and to the point.
Prepare your solution by the due date as a single ".pdf" file using the following naming convention.

"A2_COMP9016_<Surname>_<First name>_<Student Number>.pdf"

e.g.
“A2_COMP9016_OReilly_Ruairi_R123456.pdf”

3.2 Code submission

A **single** “.py’ file should be prepared containing all code for questions 1.1-1.3. Encapsulate these in three individual functions called from the end of your submitted “.py” file. **NB**: Your code will be run from a “subdir” of the AIMA libraries (I have already provided a solution for importing these libraries from a parent directory - see link - I expect you to follow these guidelines). If the datasets are small (less than 10Mb) please include these also.

Prepare your solution by the due date as a single “.py” file using the following naming convention.

“A2_COMP9016_<Surname>_<First name>_<Student Number>.py”
e.g.
“A2_COMP9016_OReilly_Ruairi_R123456.py”

3.3 Submission Penalties

- Late submissions will be penalised (-10 **points** for ≤ 1 week late, -20 **points** for > 1 week late AND ≤ 2 weeks late and **no submission accepted** after 2 weeks.
- Specified deadline is a **hard deadline** - requests for extension will not be considered (there is a formal process for individual extenuating circumstances that can be applied for via your programme coordinator e.g. medically certified illness, bereavement etc.).
- Failure to comply with submission guidelines will also result in a penalty (-10 **points**).

3.4 Submission Date

This assignment is due on Sunday of Week 10 by 21:00.