Assignment 1

COMP9418 – Advanced Topics in Statistical Machine Learning

Lecturer: Gustavo Batista

Last revision: Thursday 1st October, 2020 at 13:57

Instructions

Submission deadline: Sunday, 18th October 2020, at 18:00:00.

Late Submission Policy: The penalty is set at 20% per late day. This is ceiling penalty, so if a group is marked 60/100 and they submitted two days late, they still get 60/100.

Form of Submission: This is a group assignment. Each group can have up to **two** students. Write the names and zIDs of each student in the Jupyter notebook. **Only one member of the group should submit the assignment**.

The group should submit the solution in one single file in zip format with the name `solution.zip`. There is a maximum file size cap of 5MB, so make sure your submission does not exceed this size. The zip file should contain one Jupyter notebook file. The Jupyter notebook should have all your source code. Use markdown text to organise and explain your implementation and findings.

You are allowed to use any Python library used in the tutorial notebooks. No other library will be accepted, particularly libraries for graph and Bayesian network representation and operation. Also, you can reuse any piece of source code developed in the tutorials.

Submit your files using give. On a CSE Linux machine, type the following on the command-line:

```
$ give cs9418 ass1 solution.zip
```

Alternative, you can submit your solution via the WebCMS.

Recall the guidance regarding plagiarism in the course introduction: this applies to this assignment, and if evidence of plagiarism is detected, it will result in penalties ranging from loss of marks to suspension.

The dataset and breast cancer domain description in the Background section are from the assignment developed by Peter Lucas, Institute for Computing and Information Sciences, Radboud Universiteit.

Introduction

In this assignment, you will develop some sub-routines in Python to implement operations on Bayesian Networks. You will code an efficient independence test, learn parameters from complete data, and classify examples.

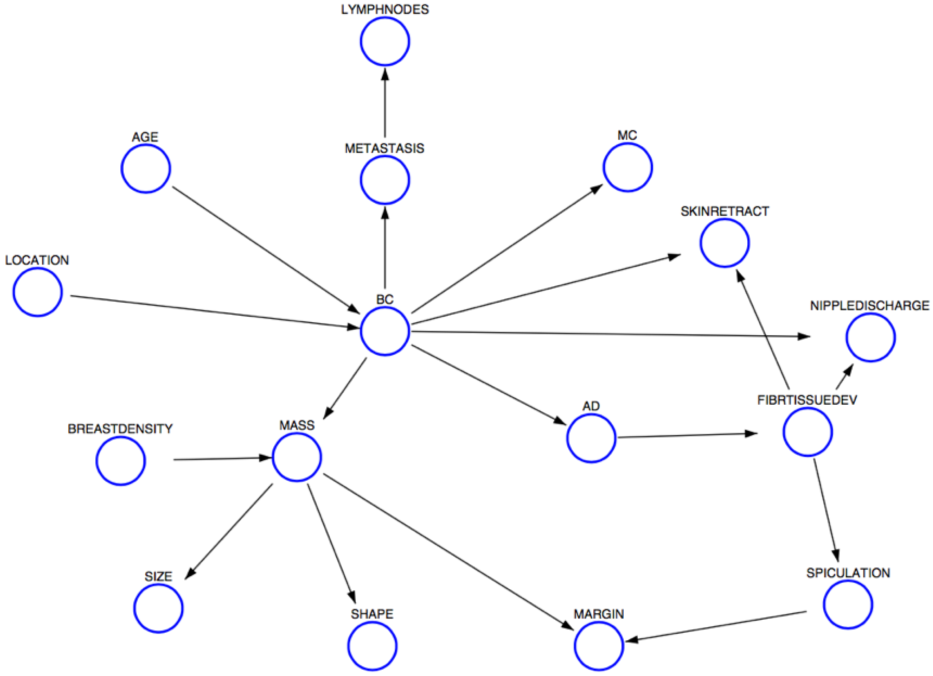
We will use a Bayesian Network for diagnosis of breast cancer. We start with some background information about the problem.

Background

Breast cancer is the most common form of cancer and the second leading cause of cancer death in women. Every 1 out of 9 women will develop breast cancer in her lifetime. Although it is not possible to say what exactly causes breast cancer, some factors may increase or change the risk for the development of breast cancer. These include age, genetic predisposition, history of breast cancer, breast density and lifestyle factors. Age, for example, is the most significant risk factor for non-hereditary breast cancer: women with age of 50 or older have a higher chance of developing breast cancer than younger women. Presence of BRCA1/2 genes leads to an increased risk of developing breast cancer irrespective of other risk factors. Furthermore, breast characteristics, such as high breast density are determining factors for breast cancer.

The primary technique used currently for detection of breast cancer is mammography, an X-ray image of the breast. It is based on the differential absorption of X-rays between the various tissue components of the breast such as fat, connective tissue, tumour tissue and calcifications. On a mammogram, radiologists can recognise breast cancer by the presence of a focal mass, architectural distortion or microcalcifications. Masses are localised findings, generally asymmetrical to the other breast, distinct from the surrounding tissues. Masses on a mammogram are characterised by several features, which help distinguish between malignant and benign (non-cancerous) masses, such as size, margin, shape. For example, a mass with irregular shape and ill-defined margin is highly suspicious for cancer, whereas a mass with round shape and well-defined margin is likely to be benign. Architectural distortion is focal disruption of the normal breast tissue pattern, which appears on a mammogram as a distortion in which surrounding breast tissues appear to be “pulled inward” into a focal point, often leading to spiculation (star-like structures). Microcalcifications are tiny bits of calcium, which may show up in clusters, or in patterns (like circles or lines) and are associated with extra cell activity in breast tissue. They can also be benign or malignant. It is also known that most of the cancers are located in the upper outer quadrant of the breast. Finally, breast cancer is characterised by several physical symptoms: nipple discharge, skin retraction, palpable lump.

Breast cancer develops in stages. The early stage is referred to as in situ (“in place”), meaning that cancer remains confined to its original location. When it has invaded the surrounding fatty tissue and possibly has spread to other organs or the lymph, so-called metastasis, it is referred to as invasive cancer. It is known that early detection of breast cancer can help improve the survival rates.



[20 Marks] Task 1 – Efficient d-separation test

In this part of the assignment, you will implement an efficient version of the d-separation algorithm. Let us start with a definition for d-separation:

Definition. Let $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ be disjoint sets of nodes in a DAG G. We will say that $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$, written dsep($\mathbf{X}$,$\mathbf{Z}$,$\mathbf{Y}$), iff every path between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ is blocked by $\mathbf{Z}$ where a path is blocked by $\mathbf{Z}$ iff there is at least one inactive triple on the path.

This definition of d-separation considers all paths connecting a node in X with a node in Y. The number of such paths can be exponential. The following algorithm provides a more efficient implementation of the test that does not require enumerating all paths.

Algorithm. Testing whether $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in a DAG G is equivalent to testing whether $\mathbf{X}$ and $\mathbf{Y}$ are disconnected in a new DAG G', which is obtained by pruning DAG G as follows:

1. We delete any leaf node W from DAG G as long as W does not belong to $X \cup Y \cup Z$. This process is repeated until no more nodes can be deleted.
2. We delete all edges outgoing from nodes in $\mathbf{Z}$.

Implement the efficient version of the d-separation algorithm in a function `d_separation(G,X,Z,Y)` that return a boolean: true if $\mathbf{X}$ is d-separated from $\mathbf{Y}$ given $\mathbf{Z}$ and false otherwise.

[10 Marks] Task 2 – Estimate Bayesian Network parameters from data

Estimating the parameters of a Bayesian Network is a relatively simple task if we have complete data. The file `bc.csv` has 20,000 complete instances, i.e., without missing values. The task is to estimate and store the conditional probability tables for each node of the graph. As we will see in more details in the Naive Bayes and Bayesian Network learning lectures, the Maximum Likelihood Estimate (MLE) for those probabilities are simply the empirical probabilities (counts) obtained from data.

Implement a function `learn_outcome_space(data)` that learns the outcome space (the valid values for each variable) from the pandas dataframe `data` and returns a dictionary `outcomeSpace` with these values.

Implement a function `learn_bayes_net(G, data, outcomeSpace)` that learns the parameters of the Bayesian Network G. This function should return a dictionary `prob_tables` with the all conditional probability tables (one for each node).

[20 Marks] Task 3 – Bayesian Network Classification

This particular Bayesian Network has a variable that plays a central role in the analysis. The variable `BC` (Breast Cancer) can assume the values `No`, `Invasive` and `InSitu`. Accurately identifying its correct value would lead to an automatic system that could help in early breast cancer diagnosis.

First, remove the variables `metastasis` and `lymphnodes` since these two variables can be understood as pieces of information derived from `BC` and they may not be available at the point when `BC` is classified.

Use the Bayesian Network to classify cases of the dataset. First, use 10-fold cross-validation to split the dataset into training and test sets. Use the function `learn_bayes_net(G, data, outcomeSpace)` to learn the Bayesian network parameters from the training set.

Design a new function `assess_bayes_net(G, prob_tables, data, outcomeSpace, class_var)` that uses the test cases in `data` to assess the performance of the Bayesian network. Implement the efficient classification procedure discussed in the lectures. Such a function should return the classifier accuracy. Compute and report the average accuracy over the ten cross-validation runs as well as the standard deviation.

[10 Marks] Task 4 – Naïve Bayes Classification

Implement a Naïve Bayes classifier. Design a new function `assess_naive_bayes(G, prob_tables, data, outcomeSpace, class_var)` to classify the cases in `data` using the log probability trick discussed in the lectures. Do 10-fold cross-validation, same as above, and return accuracy and standard deviation. Since the Naïve Bayes classifier is essentially a Bayesian network, you can call the function `learn_bayes_net(G, data, outcomeSpace)` to learn the Naïve Bayes parameters from a training set.

[20 Marks] Task 5 – Tree-augmented Naïve Bayes Classification

Similarly to the previous task, implement a Tree-augmented Naïve Bayes (TAN) classifier and evaluate your implementation in the breast cancer dataset. Design a function `learn_tan_structure(data, outcomeSpace, class_var)` to learn the TAN structure (graph) from `data` and returns such a structure.

Since the TAN classifier is also a Bayesian network, you can use the function `learn_bayes_net(G, data, outcomeSpace)` to learn the TAN parameters from a training set.

You can also use the previous designed function `assess_bayes_net(G, prob_tables, data, outcomeSpace, class_var)` to classify and assess the test cases in `data` and measure the classifier accuracy.

[20 Marks] Task 6 – Report

Write a report (**with less than 500 words**) summarising your findings in this assignment. Your report should address the following:

a. Make a summary and discussion of the experimental results (accuracy). Use plots to illustrate your results.

b. Discuss the complexity of the implemented algorithms.

Use Markdown and Latex to write your report in the Jupyter notebook. Develop some plots using Matplotlib to illustrate your results. Be mindful of the maximum number of words. Please, be concise and objective.