CS688: Graphical Models - Spring 2018

Assignment 1

Assigned: Thursday, Feb 1. Due: Thursday, Feb 15 at 11:59

Getting Started: You should complete the assignment using your own installation of Python 2.7. The only module you are permitted to use in your implementations is Numpy. To get started with the code portions of the assignment, download the assignment archive from Moodle and unzip the file. The data files for this assignment are in the `data` directory. Code templates are in the `code` directory. The README file contains additional descriptions of the download. **Note**: The autograder uses numpy 1.14. Please use this version of numpy. Note that the autograder environment will fail if you attempt to load other packages.

Deliverables: This assignment has two types of deliverables: a report and code files.

- **Report:** The solution report will give your answers to the homework questions. Items that you should include in your report are marked with **(report)**. The maximum length of the report is 5 pages in 11 point font, including all figures and tables. You can use any software to create your report, but your report must be submitted in PDF format. You will upload the PDF of your report to Gradescope under `HW01-Report` for grading. It is strongly recommended that you typeset your report. To assist with this if you wish to use Latex, the Latex source of the handout is also included in the homework archive.
- **Code:** The second deliverable is your code. Items that you should include in your code are marked with **(code)**. Your code must be Python 2.7 (no iPython notebooks, other formats, or code from other versions of Python). You will upload a zip file (not rar, bz2 or other compressed format) containing all of your code to Gradescope under `HW01-Programming` for autograding. When unzipped, your zip file should produce a directory called `code`. If your zip file has the wrong structure, the autograder may fail to run.

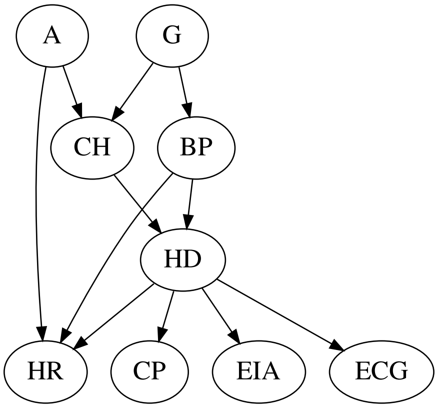
Academic Honesty Statement: Copying solutions from external sources (books, web pages, etc.) or other students is considered cheating. Sharing your solutions with other students is also considered cheating. Collaboration indistinguishable from copying is a violation of the course's collaboration policy and will be treated as cheating. Any detected cheating will result in a grade of 0 on the assignment for all students involved, and potentially a grade of F in the course.

Introduction: In this assignment, you will experiment with different aspects of modeling, learning, and applying a Bayesian network to answer probability queries. This assignment focuses on the heart disease diagnosis domain and uses part of a real clinical data set.

Data Set: The data set consists of 9 variables as described below. The number of each variable corresponds to it's column number in the data set files. There are five sets of training and test data files in standard comma-separated-value (CSV) format. The files are named *data-train-i.txt* and *data-test-i.txt* for i from 1 to 5.

Number	Name	Description	Values
1	A	Age	1:< 45, 2: $45 - 55$, 3:≥ 55
2	G	Gender	1:Female, 2:Male;
3	CP	Chest Pain	1:Typical, 2:Atypical, 3:Non-Anginal, 4:None
4	BP	Blood Pressure	1:Low, 2:High
5	CH	Cholesterol	1:Low, 2:High
6	ECG	Electrocardiograph	1:Normal, 2:Abnormal
7	HR	Exercise Heart Rate	1:Low, 2:High
8	EIA	Exercise Induced Angina	1:No, 2:Yes
9	HD	Heart Disease	1:No, 2:Yes

Model: We will consider applying a Bayesian network with the following structure to the data set.



1. (*10 points*) **Factorization** : Write down the factorization of the Bayesian network joint distribution implied by the structure shown above. **(report)**

2. (*10 points*) **Likelihood Function:** Using the notation for the parameters of CPTs introduced in Lecture 4 (ie: $P_\theta(HD = hd|CH = ch, BP = bp) = \theta^{HD}_{hd|ch,bp}$), write down the log likelihood of the Bayesian network model as a function of the parameters θ given N data cases. **(report)**

3. (*10 points*) **Maximum Likelihood Estimates:** Using the notation for the parameters of CPTs introduced in Lecture 4, derive the maximum likelihood estimate for the parameter $\theta^{HR}_{L|1,H,Y}$ starting from the log likelihood function. Show all of your work. **(report)**

4. (*15 points*) **CPT Data Structures** In this question, you will describe and implement conditional probability table data structures for the Bayesian network shown above.

4.1. (*5 pts*) To implement the network shown above, you will need to choose a data structure to represent conditional probability tables. There are many possible choices including classes of your own design. As your answer to this question, explain what data structure you chose to represent CPTs and how it represents CPTs with different numbers of parents and different variable cardinalities. If you implement a custom class, describe its member variables and methods. If you use an existing data structure (lists, array, dictionary, etc.), describe how. (**report**)

4.2. (*5 pts*) In the `BayesNet` class contained in `code/bn.py`, implement the `__init__` method. This method should create the set of CPTs needed to represent the above model as member variables according to your approach described in 4.1. You should initialize each CPT to the uniform distribution over the target variable for each setting of the parent variables. (**code**)

4.3. (*5 pts*) In the `BayesNet` class contained in `code/bn.py`, implement the methods `get` and `set`. These methods provide basic functions to get the current value of a specific probability from a CPT member variable and to set a specific probability in a CPT member variable in a way that is independent of your CPT implementation. (**code**)

5. (*15 points*) **Learning:** In this question, you will describe and implement learning for the Bayesian network shown above.

5.1. (*5 pts*) To implement maximum likelihood learning for a Bayesian network model, the parameters of each CPT must be estimated from data. This can be accomplished by writing unique code for every CPT in the model. This is conceptually simple, but highly redundant and error prone. It is also possible to write a single general function that can estimate the parameters for any CPT. This is conceptually more complex, but provides much more compact code. As your answer to this question, describe your approach to implementing learning. (**report**)

5.2. (*10 pts*) In the `BayesNet` class contained in `code/bn.py`, implement the `fit` function according to the approach you described in part 1. The `fit` function must implement maximum likelihood parameter estimation for all CPTs in the model. (**code**)

6. (*10 points*) **Probability Queries:** In this question, you will derive and implement solutions to two different probability queries.

6.1. (*5 pts*) For each of the two queries listed below, show how the query can be expressed in terms of the factorized joint distribution for the Bayesian network. Simplify the expressions wherever possible using the conditional independence properties of the network structure. Show your work. (**report**)

(a) $P(CH|A = 2, G = M, CP = None, BP = L, ECG = Normal, HR = L, EIA = No, HD = No)$
(b) $P(BP|A = 2, CP = Typical, CH = H, ECG = Normal, HR = H, EIA = Yes, HD = No)$

6.2. (*5 pts*) In `code/queries.py`, implement the methods `query_5_a` and `query_5_b`. These two functions take an instance of the BayesNet class as input and should compute the query distributions for queries (a) and (b). (**code**)

7. (*15 points*) **Classification:** In this question, we will assess the ability of the model to correctly predict the occurrence of heart disease given the values of all of the other variables in the network.

7.1. (*5 pts*) Write down the probability distribution over the heart disease variable (HD) given the remaining variables for the Bayesian Network model shown above. Simplify the result using the conditional independence properties of the network **(report)**.

7.2. (*5 pts*) Implement the `predict_hd` method of the BayesNet class in `code/bn.py`. This method will predict if a patient has heart disease or not using the probability of heart disease given the remaining variables as derived in 7.1. Your method should predict Yes (2) if the probability of heart disease if greater than 0.5 and should predict No (1) otherwise **(code)**.

7.3. (*5 pts*) We will follow a standard five-fold cross-validation protocol to assess the performance of classifier derived from the model in 7.2. For each training file i, use your Bayesian network implementation to learn the parameters of the model, then make predictions for the heart disease variable for each data case n in test file i using the `predict_hd` method. For each test file i, compute the prediction accuracy A_i as the number of cases correctly predicted divided by the total number of cases. Lastly, compute the mean prediction accuracy over the five test files (the average of A_1 to A_5) and the standard deviation of the prediction accuracy over the five test files (the standard deviation of A_1 to A_5). Report the mean and the standard deviation of the prediction accuracy that you find **(report)**. Add your code to **accuracy.py** **(code)**.

8. (*15 points*) **Modeling:** In this question, you will design, implement and evaluate your own network structure for the heart disease domain.

8.1. (*2 pts*) Provide a figure showing the graphical model for your network **(report)**.

8.2. (*3 pts*) Write down the factorization for your network **(report)**.

8.3. (*3 pts*) Briefly describe some of the choices that went into the design of your network **(report)**.

8.4. (*5 pts*) Implement your network in `code/bn_custom.py`. The API for this class is identical to that of `code/bn.py`. You must support the `fit`, `get`, `set`, and `predict_hd` methods **(code)**.

8.5. (*2 pts*) As in the previous question, apply the five-fold cross-validation protocol to assess and report the mean prediction accuracy of your model, along with its standard deviation. **(report)**