CSC 665: Artificial Intelligence

Homework 3

By turning in this assignment, I agree to abide by SFSU's academic integrity code and declare that all of my solutions are my own work.

1 Probability

a. (7 points) An urn contains w white balls and b black balls. Two balls are drawn, one after the other, without replacement. What is the probability that the first ball is white? What is the probability that the second ball is white?

b. (7 points) You meet Alice. Alice tells you she has two brothers, Bob and Charlie. What is the probability that Alice is older than Charlie?

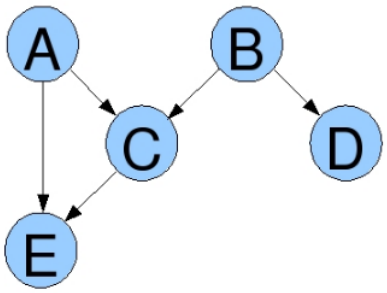
Alice tells you that she is older than Bob. Now what is the probability that Alice is older than Charlie?

c. (7 points) The inhabitants of an island tell the truth one third of the time. They lie with probability 2/3. On an occasion, after one of them made a statement, you ask another, "was that statement true?" and he says "yes." What is the probability that the statement was indeed true?

d. (7 points) A bag contains one ball, known to be either white or black with equal probability. A white ball is put in, the bag is shaken, and a ball is drawn out, which proves to be white. Now what is the probability that the remaining ball is white? (Notice that the state of the bag, after the operations, is exactly identical to its state before.)

2 Bayesian networks

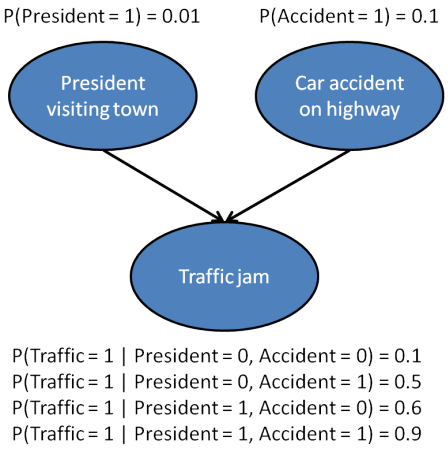
a. (8 points) Given the following Bayesian network, write down a factorization of the joint probability distribution $P(A, B, C, D, E)$ as a product of local conditional distributions, one for each random variable.



b. (8 points) For the same Bayesian network above, list all the random variables that are independent of A, assuming that none of the variables have been observed.

c. (8 points) For the same Bayesian network above, assume that E is now observed. List all pairs of random variables (not including E) that are conditionally independent given E.

d. (8 points) Consider the following model for traffic jams in a small town, which can be caused by either a car accident or by a visit from the president (and the accompanying security motorcade).



Compute $P(\text{Accident} = 1 \mid \text{Traffic} = 1)$ and $P(\text{Accident} = 1 \mid \text{Traffic} = 1, \text{President} = 1)$.

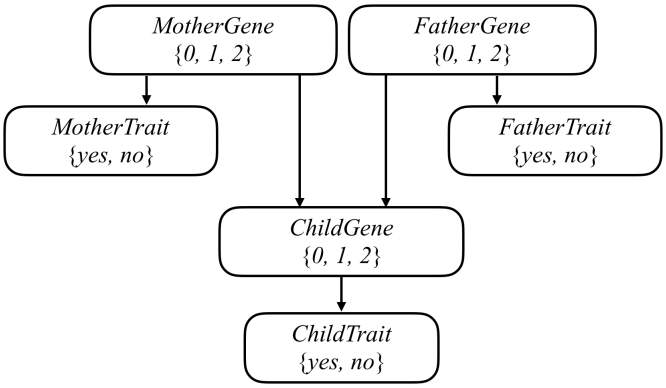
3 Genetics

Background

Mutated versions of the GJB2 gene are one of the leading causes of hearing impairment in newborns. Each person carries two copies of the gene, so each person has the potential to possess either 0, 1, or 2 copies of the hearing impairment version of GJB2. Unless a person undergoes genetic testing though, it's not so easy to know how many copies of mutated GJB2 a person has. This is some "hidden state": information that has an effect that we can observe (hearing impairment), but that we don't necessarily directly know. After all, some people might have 1 or 2 copies of mutated GJB2 but not exhibit hearing impairment, while others might have no copies of mutated GJB2 yet still exhibit hearing impairment.

Every child inherits one copy of the GJB2 gene from each of their parents. If a parent has two copies of the mutated gene, then they will pass the mutated gene on to the child; if a parent has no copies of the mutated gene, then they will not pass the mutated gene on to the child; and if a parent has one copy of the mutated gene, then the gene is passed on to the child with probability 0.5. After a gene is passed on though, it has some probability of undergoing additional mutation: changing from a version of the gene that causes hearing impairment to a version that doesn't, or vice versa.

We can attempt to model all of these relationships by forming a Bayesian network of all the relevant variables, as in the one below, which considers a family of two parents and a single child.
Each person in the family has a Gene random variable representing how many copies of a particular gene (e.g., the hearing impairment version of GJB2) a person has. The possible values of this random variable are 0, 1, or 2. Each person in the family also has a Trait random variable, which is yes or no depending on whether that person expresses a trait (e.g., hearing impairment) based on that gene. There's an arrow from each person's Gene variable to their Trait variable to encode the fact that a person's genes affect the probability that they have a particular trait. Meanwhile, there's also an arrow from both the mother's and father's Gene random variables to their child's Gene random variable: the child's genes are dependent on the genes of their parents.



Your task in this project is to use this model to make inferences about a population. Given information about people, who their parents are, and whether they have a particular observable trait (e.g. hearing loss) caused by a given gene, you will infer the probability distribution for each person's genes, as well as the probability distribution for whether any person will exhibit the trait in question.

Code

Take a look at one of the sample data sets in the `data/` directory by opening up `data/family0.csv`. Notice that the first row defines the columns for this CSV file: name, mother, father, and trait. The next row indicates that Harry has Lily as a mother, James as a father, and the empty cell for trait means we don't know whether Harry has the trait or not. James, meanwhile, has no parents listed in the our data set (as indicated by the empty cells for mother and father), and does exhibit the trait (as indicated by the 1 in the trait cell). Lily, on the other hand, also has no parents listed in the data set, but does not exhibit the trait (as indicated by the 0 in the trait cell).

Open up `heredity.py` and take a look first at the definition of `PROBS`, which is a dictionary containing a number of constants representing probabilities of various events. All of these events have to do with how many copies of a particular gene a person has (hereafter referred to as simply "the gene"), and whether a person exhibits a particular trait (hereafter referred to as "the trait") based on that gene. The data here is loosely based on the probabilities for the hearing impairment version of the GJB2 gene and the hearing impairment trait.

First, `PROBS["gene"]` represents the unconditional probability distribution over the gene (i.e., the probability if we know nothing about that person's parents). Based on the data in the distribution code, it would seem that in the population, there's a 1% chance of having 2 copies of the gene, a 3% chance of having 1 copy of the gene, and a 96% chance of having 0 copies of the gene.

Next, `PROBS["trait"]` represents the conditional probability that a person exhibits a trait (like hearing impairment). This is actually three different probability distributions: one for each possible value for gene. So `PROBS["trait"][2]` is the probability distribution that a person has the trait given that they have two versions of the gene: in this case, they have a 65% chance of exhibiting the trait, and a 35% chance of not exhibiting the trait. Meanwhile, if a person has 0 copies of the gene, they have a 1% chance of exhibiting the trait, and a 99% chance of not exhibiting the trait.

Finally, `PROBS["mutation"]` is the probability that a gene mutates from being the gene in question to not being that gene, and vice versa. If a mother has two versions of the gene, for example, and therefore passes one on to her child, there's a 1% chance it mutates into not being the target gene anymore. Conversely, if a mother has no versions of the gene, and therefore does not pass it onto her child, there's a 1% chance it mutates into being the target gene. It's therefore possible that even if neither parent has any copies of the gene in question, their child might have 1 or even 2 copies of the gene.

Ultimately, the probabilities you calculate will be based on these values in `PROBS`.

Now, take a look at the `main` function. The function first loads data from a file into a dictionary `people`, which maps each person's name to another dictionary containing information about them, including their name, their mother (if one is listed in the data set), their father (if one is listed in the data set), and whether they are observed to have the trait in question (`True` if they do, `False` if they don't, and `None` if we don't know).

Next, `main` defines a dictionary of probabilities, with all probabilities initially set to 0. This is ultimately what your program will compute: for each person, you will calculate the probability distribution over how many of copies of the gene they have, as well as whether they have the trait or not. `probabilities["Harry"]["gene"][1]`, for example, will be the probability that Harry has 1 copy of the gene, and `probabilities["Lily"]["trait"][False]` will be the probability that Lily does not exhibit the trait.

Task

Ultimately, we're looking to calculate these probabilities based on some evidence: given that we know certain people do or do not exhibit the trait, we'd like to determine these probabilities. Recall from lecture that we can calculate a conditional probability by summing up all of the joint probabilities that satisfy the evidence, and then normalizing those probabilities so that they sum to 1. Your task in this project is to implement three functions to do just that: `joint_probability` to compute a joint probability, `update` to add the newly computed joint probability to the existing probability distribution, and then `normalize` to ensure all probability distributions sum to 1 at the end.

a. (20 points) Implement `joint_probability`, which takes as input a dictionary of people, along with data about who has how many copies of each of the genes, and who exhibits the trait. The function should return the joint probability of all of those events taking place.

b. (10 points) Implement `update`, which adds a new joint probability to the existing probability distributions in `probabilities`.

c. (10 points) Implement `normalize`, which updates a dictionary of probabilities such that each probability distribution is normalized (i.e., sums to 1, with relative proportions the same).

You should not modify anything else in `heredity.py` other than these three functions, although you may write additional functions and/or import other Python standard library modules.

Example

To help you think about how to calculate joint probabilities, here's an example. Consider the following value for `people`:

```
{
  'Harry': {'name': 'Harry', 'mother': 'Lily', 'father': 'James', 'trait': None},
  'James': {'name': 'James', 'mother': None, 'father': None, 'trait': True},
  'Lily': {'name': 'Lily', 'mother': None, 'father': None, 'trait': False}
}
```

We will here show the calculation of `joint_probability(people, {"Harry"}, {"James"}, {"James"})`. This represents the probability that Lily has 0 copies of the gene and does not have the trait, Harry has 1 copy of the gene and does not have the trait, and James has 2 copies of the gene and does have the trait.

We start with Lily (the order that we consider people does not matter, so long as we multiply the correct values together, since multiplication is commutative). Lily has 0 copies of the gene with probability 0.96 (this is `PROBS["gene"][0]`). Given that she has 0 copies of the gene, she doesn't have the trait with

probability 0.99 (this is `PROBS["trait"][0][False]`). Thus, the probability that she has 0 copies of the gene and she doesn't have the trait is $0.96 \cdot 0.99 = 0.9504$.

Next, we consider James. James has 2 copies of the gene with probability 0.01 (this is `PROBS["gene"][2]`). Given that he has 2 copies of the gene, the probability that he does have the trait is 0.65. Thus, the probability that he has 2 copies of the gene and he does have the trait is $0.01 \cdot 0.65 = 0.0065$.

Finally, we consider Harry. What's the probability that Harry has 1 copy of the gene? There are two ways this can happen. Either he gets the gene from his mother and not his father, or he gets the gene from his father and not his mother. His mother Lily has 0 copies of the gene, so Harry will get the gene from his mother with probability 0.01 (this is `PROBS["mutation"]`), since the only way to get the gene from his mother is if it mutated; conversely, Harry will not get the gene from his mother with probability 0.99. His father James has 2 copies of the gene, so Harry will get the gene from his father with probability 0.99 (this is $1 -$ `PROBS["mutation"]`), but will get the gene from his mother with probability 0.01 (the chance of a mutation). Both of these cases can be added together to get $0.99 \cdot 0.99 + 0.01 \cdot 0.01 = 0.9802$, the probability that Harry has 1 copy of the gene.

Given that Harry has 1 copy of the gene, the probability that he does not have the trait is 0.44 (this is `PROBS["trait"][1][False]`). So the probability that Harry has 1 copy of the gene and does not have the trait is $0.9802 \cdot 0.44 = 0.431288$.

Therefore, the entire joint probability is just the result of multiplying all of these values for each of the three people: $0.9504 \cdot 0.0065 \cdot 0.431288 = 0.0026643247488$.

Submission

Submission is done on Canvas. You should submit two files: one containing your solutions to the written problems, and one for the coding problem.

- Submit your written solutions in a single PDF file with your name at the top. Make sure to clearly indicate the number and letter of the problem corresponding to each solution. It is okay to hand-write your solutions and then scan them into a PDF, but *only if your handwriting is legible.*
- Submit your coding solution in the provided `heredity.py` file. Do not modify the name of this file after downloading it from the course website.