CPSC 540 Assignment 4 (due March 20)

Graphical Models and Paper Review

1 Markov Chains

1.1 Sampling, Inference, and Decoding

The function *example_markovChain.m* loads the initial state probabilities and transition probabilities for three Markov chain models on d binary variables,

$$p(x_1, x_2, \ldots, x_d) = p(x_1) \prod_{j=2}^{d} p(x_j | x_{j-1}).$$

It then tries to find the optimal decoding (the most likely assignment to the variables $\{x_1, x_2, \ldots, x_d\}$) in each of the three chains. In the demo, decoding is done by enumerating all possible assignments to the variables. This works for the first two chains as they only have 4 variables, but is too slow on the last chain because it has 30 variables. In this question you'll explore two ways to estimate the marginals in third Markov chain and two ways to estimate the most-probable sequence.

1. Write a function, *sampleAncestral.m*, that uses ancestral sampling to sample sequence x. Hand in this code and report all the univariate marginal probabilities using a Monte Carlo estimate based on 10000 samples.

2. Write a function, *marginalCK.m*, that uses the CK equations to compute the exact univariate marginals. Hand in this code and report all exact univariate marginals.

3. Write a function, *marginalDecode.m*, that returns the sequence of states x_j that maximize the marginal probability $p(x_j)$ (for each j). Hand in this code and report the sequence of most likely states.

4. Write a function, *viterbiDecode.m*, that implements the Viterbi decoding algorithm for Markov chains. Hand in this code and report the optimal decoding of the third Markov chain.

Hint: for parts 2-4, you can use a 2 by d matrix M to represent the dynamic programming table, and for part 4 you can use another matrix B containing the argmax values that lead to each entry in the table.

1.2 Conditioning

The long sequence from the previous question usually starts with state 1 and most of the time ends in state 2. In this question you'll consider conditioning on these events not happening. First, compute the following quantities which can be done using your functions from the previous question:

1. Report all the univariate conditional probabilities $p(x_j | x_1 = 2)$ obtained using a Monte Carlo estimate based on 10000 samples.

2. Report all the exact univariate conditionals $p(x_j | x_1 = 2)$.

3. Report the sequence beginning with $x_1 = 2$ that has the highest probability.

4. Report the sequence ending with $x_d = 1$ that has the highest probability.

Hint: these conditions can be done by changing the input to the functions from the previous question.

Next consider the following cases (which require implementing an extra rejection step or backward phase):

5. Report all the univariate conditional probabilities $p(x_j|x_d = 1)$ obtained using a Monte Carlo estimate based on 10000 samples and rejection sampling. Also report the number of samples accepted among the 10000 samples.

6. Write a function, *sampleBackwards.m* that uses backwards sampling to sample sequences where $x_d = 1$). Hand in this code and report all the univariate conditional probabilities $p(x_j|x_d = 1)$ obtained using a Monte Carlo estimate based on 10000 samples.

7. Write a function, *forwardBackwards.m* that is able compute all exact univariate conditionals $p(x_j|x_d = 1)$ in $O(dk^2)$. Hand in the code and report all the exact univariate conditionals $p(x_j|x_d = 1)$.

1.3   1D Linear-Gaussian Markov Chains

Consider a continuous-state Markov chain where the initial distribution is given by

$$x_0 \sim \mathcal{N}(m_0, v_0^2),$$

and the transition distributions for $j > 1$ are given by

$$x_j|x_{j-1} \sim \mathcal{N}(w_j x_{j-1} + m_j, v_j^2).$$

This model could be used to model an object moving through $\mathbb{R}$.[1] Because of the Gaussian assumptions, this defines a joint Gaussian distribution over the variables while the marginal distributions are also Gaussian. For a generic $j > 1$, derive the form of the marginal distribution of x_j, expressing the marginal parameters μ_j and σ_j recursively in terms of μ_{j-1} and σ_{j-1}.

Hint: You can use Theorem 4.4.1 of Murphy's book.

2   Directed Acyclic Graphical Models

2.1   D-Separation

Consider a directed acyclic graphical (DAG) model with the following graph structure:


---
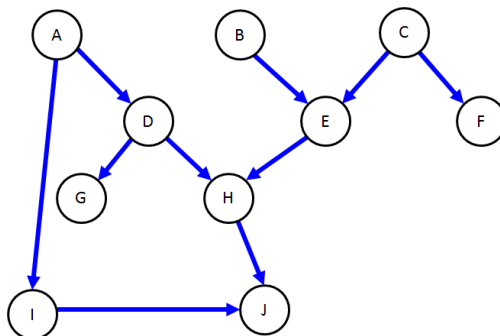
[1]In practical applications like object tracking, we typically have that the states x_j are 2- or 3-dimensions so we model an object like a submarine or an airplane moving through space.

Assuming that the conditional independence properties are faithful to the graph, using d-separation briefly explain why the following are true or false:

1. $B \perp F$.

2. $B \perp F \mid A$.

3. $B \perp F \mid C$.

4. $B \perp F \mid E$.

5. $B \perp F \mid I$.

6. $B \perp F \mid J$.

7. $B \perp F \mid C, E$.

2.2 Exact Inference

While DAGs can be used as a visual representation of independence assumptions, they can also be used to simplify computations.This question will give you practice using the basic properties which allow efficient computations in graphical models. Consider the DAG model below, for distinguishing between different causes of shortness-of-breath (dyspnoea) and the causes of an abnormal lung x-ray, while modelling potential causes of these diseases too (whether the person is a smoker or had a 'visit' to a country with a high degree of tuberculosis).
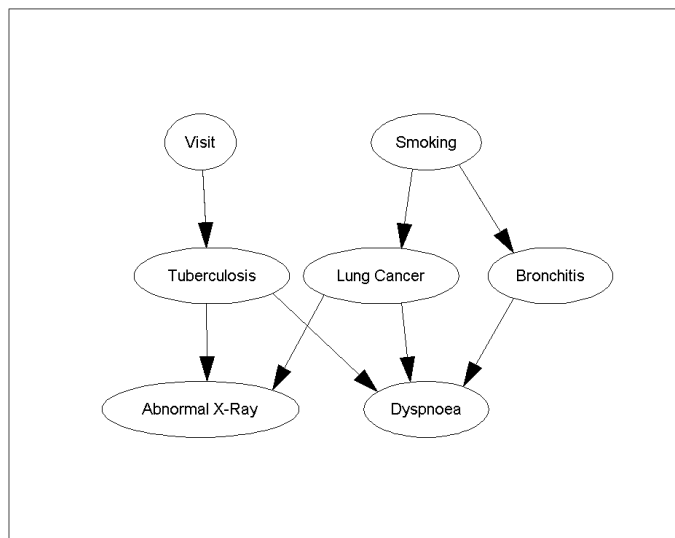
For this question, let's assume that we use the following parameterization of the network:

$$\text{Visit}$$
$$p(V = 1) = 0.01$$
$$\text{Smoking}$$
$$p(S = 1) = 0.2$$
$$\text{Tuberculosis}$$
$$p(T = 1 | V = 1) = 0.05$$
$$p(T = 1 | V = 0) = 0.01$$
$$\text{Lung Cancer}$$
$$p(L = 1 | S = 1) = 0.10$$
$$p(L = 1 | S = 0) = 0.01$$
$$\text{Bronchitis}$$
$$p(B = 1 | S = 1) = 0.60$$
$$p(B = 1 | S = 0) = 0.30$$
$$\text{Abnormal X-Ray}$$
$$p(X = 1 | T = 1, L = 1) = 1.00$$
$$p(X = 1 | T = 1, L = 0) = 0.98$$
$$p(X = 1 | T = 0, L = 1) = 0.9$$
$$p(X = 1 | T = 0, L = 0) = 0.05$$

$$\text{Dyspnoea}$$
$$p(D = 1 | T = 1, L = 1, B = 1) = 0.90$$
$$p(D = 1 | T = 1, L = 1, B = 0) = 0.70$$
$$p(D = 1 | T = 1, L = 0, B = 1) = 0.85$$
$$p(D = 1 | T = 1, L = 0, B = 0) = 0.65$$
$$p(D = 1 | T = 0, L = 1, B = 1) = 0.82$$
$$p(D = 1 | T = 0, L = 1, B = 0) = 0.60$$
$$p(D = 1 | T = 0, L = 0, B = 1) = 0.80$$
$$p(D = 1 | T = 0, L = 0, B = 0) = 0.10$$

Compute the following quantities (hints are given on the right, and these will be easier to do in order and if

you use conditional independence properties to simplify the calculations):

0. $p(S = 1)$ (marginal of root node; can read from table)

1. $p(S = 0)$ (negation of marginal of root node; use sum to one constraint)

2. $p(L = 1 | S = 1)$ (conditional of child node given parents; can be read from table)

3. $p(L = 1)$ (marginal of child node; marginalize over parent)

4. $p(X = 1 | T = 1, L = 1)$ (conditional of child given parents; can be read from table)

5. $p(X = 1 | T = 1)$ (conditional of child with missing parent; marginalize over missing parent)

6. $p(X = 1 | T = 1, S = 1)$ (conditional of child given parent and grand-parent, marginalize over missing parent)

7. $p(X = 1)$ (marginal of leaf node; marginalize over parents and use independence to simplify)

8. $p(T = 1 | X = 1)$ (conditional of parent given child; use Bayes rule)

9. $p(T = 1 | L = 1)$ (conditional of parent given co-parent; use independence and then marginal)

10. $p(T = 1 | X = 1, L = 1)$ (conditional of parent given child and co-parent; use Bayes rule)

2.3   Inpainting

The function *example_fil.m* loads a variant of the MNIST dataset. It contains all the training images but the test images are missing their bottom half. Running this function fits an independent Bernoulli model to the training set, and then shows the result of applying the density model to "fill in" four random test examples. It performs pretty badly because the independent model can't condition on the known top-half of the images.

1. Make a variant of the demo where you fit an inhomogeneous Markov chain to each image column. Hand in your code and an example of using this model to fill in 4 random test images.

2. Make a variant of the demo where you fit a directed acyclic graphical model to the data, using general discrete conditional probabilities and where the parents of pixel (i, j) are the other 8 pixels in the region $(i - 2 : i, j - 2 : j)$. Hand in your code and an example of using this model to fill in 4 random test images.

3. Consider using more than 8 pixels are parents in the above model, such as the 15 pixels in the region $(i - 3 : i, j - 3 : j)$. If you do this, the code will often place white pixels in the bottom right corner of the image even though no training example has a white pixel there. Why would it do this?

4. Make a variant of the demo where you fit a sigmoid belief network to the data, where the parents of pixel (i, j) are the other pixels in the region $(1 : i, 1 : j)$. Hand in your code and an example of using this model to fill in 4 random test images.

Hints: For parts 2 and 3, you may find it helpful to make an m by m cell array called *models* where element (i, j) contains the model for pixel (i, j). For parts 2 and 3 the size of the dataset also mean you will probably need to vectorize your computation. The functions *permute* and *reshape* will help you, making a sparse version of X with *sparse* can also speed up many operations. For part 2, you can use *binaryTabular.m* to fit the discrete conditional distribution and sample from it (a reasonable value of α is 1). For part 3, you can use *logisticL2.m* to fit logistic regression models and sample from them (a reasonable value of λ is 1). Note that *logisticL2.m* uses a $\{-1, 1\}$ encoding of y while *binaryTabular.m* uses a $\{0, 1\}$ encoding (both support sparse X).

3 Paper Review

3.1 Finding Relevant Papers

To help you make progress on your project, for this part you should hand in a list of 10 academic papers related to your current project topic. Finding related work is often one of the first steps towards getting a new project started, and it gives you an idea of what has (and has not) been explored. Some strategies for finding related papers are:

1. Use Google: try the keywords you think are most relevant. Asking people in your lab (or related labs) for references is also often a good starting point.

2. Once you have found a few related papers, read their introduction section to find references that these papers think are worth mentioning.

3. Once you have found a few related papers, use Google Scholar to look through the list of references that are *citing* these papers. You may have to do some sifting if there are a lot of citations. Reasonable criteria to sift through large reference lists include looking for the ones with the most citations or focusing on the more recent ones (then returning to Step 2 to find the more-relevant older references).

For this question you only need to provide a list, but in Assignment 5 you will have to do a survey of 10 papers. So it's worth trying to identify papers that are both relevant and important at this point. For some types of projects it will be easier to find papers than others. If you are having trouble, post on Piazza.

Although the papers do not need to all be machine learning papers, the course project does need to be related to machine learning in some way, so at least a subset of the papers should be machine learning papers. Here is a rough guide to some of the most reputable places to where you see machine learning works published:

- The International Conference on Machine Learning (ICML) and the conference on Advances in Neural Information Processing (NIPS) are the top places to publish machine learning work. The Journal of Machine Learning Research (JMLR) is the top journal, although in this field conference publications are usually viewed as more prestigious.

- Other good venues include AISTATS (emphasis on statistics), UAI (emphasis on graphical models), COLT (emphasis on theory), ICLR (emphasis on deep learning), ECML-PKDD (European version of ICML), CVPR and ICCV/ECCV (emphasis on computer vision), ACL and EMNLP (emphasis on language), KDD (emphasis on data mining), AAAI/IJCAI (emphasis on AI more broadly), JRSSB and Annals of Stats (emphasis on statistics more broadly), and Science and Nature (emphasis on science more broadly).

3.2 Paper Review

Among your list of 10 papers, choose one paper and write a review of this paper. It makes sense to choose one of the most closely-related to your project or one of the most important-looking papers. The review should have two parts:

1. A short summary of the contributions of the paper. Say what problem the paper is addressing, why this is an important problems, what is proposed, and how it is being evaluated.

2. A list of strengths and weaknesses of the paper. For ideas of what issues to discuss, see the JMLR guidelines for reviewers: `http://www.jmlr.org/reviewer-guide.html`

Note that you should include a non-empty list of strengths *and* weaknesses. Many students when doing their first reviews focus either purely on strengths or purely on weaknesses. It's important to recognize that all papers have weaknesses or limitations (even ones written by famous people or that are published

in impressive places or that proved to be historically important) and all papers have strengths or at least motivation (the authors must have thought it was worth writing for some reason).