CS688: Graphical Models - Spring 2014

Assignment 5

Assigned: Tuesday, Apr 15. Due: Tuesday, Apr 29 at 2:30

General Instructions: Submit a report with the answers to each question at the start of class on the date the assignment is due. You are encouraged to typeset your solutions. To help you get started, the full LaTeXsource of the assignment is included with the assignment materials. For your assignment to be considered "on time", you must upload a zip file containing all of your code to Moodle by the due date. Make sure the code is sufficiently well documented that it's easy to tell what it's doing. You may use any programming language you like. Make sure to list in your report any outside references you consulted (books, articles, web pages, etc.) and any students you collaborated with.

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. 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 develop a model, learning algorithm, and inference algorithm for a text modeling problem. Unlike the assignments to this point, you are free to select any probabilistic graphical model of your choosing and pair it with any learning and inference algorithms. You are free to use models and algorithms that you have already implemented for this class. You can not use models or algorithms from external libraries or packages.

Data Set and Task: For this assignment, you will use a data set consisting of word occurrences in newsgroup posts. Each data case $\mathbf{x}_n$ consists of a newsgroup post represented as a 104 element binary vector. The first 100 elements each correspond to the presence/absence of a word. If word w occurred in post n, when $x_{nw} = 1$, otherwise $x_{nw} = 0$. The last four columns indicate which of the four top-level news groups the post belongs to. Each of these columns is also binary. The task is to learn a model on the training data, and to use this model to infer marginal probability distributions over different combinations of unobserved variables in the test data set. The training data is in the file *train.txt*. The test data is in the file *test.txt*. The file *groups.txt* lists the name of the newsgroups. The file *words.txt* lists the 100 words in the same order that they appear in the data vectors.

1. (*30 points*) **Modeling:** To begin, you will need to choose a model for this task. Note that you can design a custom probabilistic graphical model for this problem, or use any of the models we have used in past assignments. Once you have made your choice, answer the following questions.

(a) [15] Write a paragraph describing the model you have selected and arguing why you think it is a good choice for this task. Clearly describe the criteria you applied in selecting your model.

(b) [10] Give the mathematical form of the joint distribution of your model.

(c) [5] How many parameters does your model contain? Does your model contain any hyper-parameters (regularization constants, etc.)?

2. (*40 points*) **Learning:** Before you can use your model for inference, you must select and implement a learning approach for your model. Note that you can use a custom learning algorithm for your model, or adapt any appropriate learning algorithm we have used in past assignments. Once you have made your choice, answer the following questions.

(a) [15] Write a paragraph describing the algorithm you have selected and arguing why you think it is a good choice. Clearly describe the criteria you applied in selecting your algorithm. Note that for some models there will only be one logical learning algorithm choice. If your model contains hyper-parameters, discuss how you will select their values.

(b) [15] Give a detailed pseudo code description of your learning algorithm. Provide sufficient detail that someone who has taken the course would have a reasonable chance of correctly implementing your algorithm based on your pseudo code.

(c) [5] Give a complexity analysis of your learning algorithm focusing on its dependence on the number of data cases N, and the number of variables M. For iterative algorithms, analyze the complexity of a single iteration.

(d) [5] Train your model and supply at least one plot relating to training. For non-iterative methods, consider plotting the running time of your training algorithm as a function of the number of training cases. For iterative training algorithms, supplying a plot relating to convergence of the algorithm across iterations would be appropriate.

3. (*20 points*) **Inference:** Before you can apply your learned model to the task of interest, you must select and implement an inference method for your model. Your algorithm must be able to produce single variable marginals over multiple unobserved variables at test time. Note that you can use a custom inference algorithm for your model, or adapt any appropriate inference algorithm we have used in past assignments. Once you have made your choice, answer the following questions.

(a) [10] Write a paragraph describing the algorithm you have selected and arguing why you think it is a good choice. Clearly describe the criteria you applied in selecting your algorithm.

(b) [10] Give a detailed pseudo code description of your inference algorithm. Provide sufficient detail that someone who has taken the course would have a reasonable chance of correctly implementing your algorithm based on your pseudo code.

4. (*10 points*) **Prediction:** Once your model is trained, you will apply it to produce marginal probabilities over unobserved values in a variety of test cases. The test cases are defined in the file *text.txt*. Observed variables take the values 0 or 1. Unobserved variables are coded as -1. As your answer to this question, you will compute the log marginal probability $\log P(X_{ni} = 1)$ for each test case n and each unobserved variable i within that test case and write them to a text file. You should output the log marginal probabilities for test case n on line n of the file. You should list the log marginal probabilities separated by spaces in the order that the corresponding variables appear in the original data. Name this file *marginals.txt* and include it with the code you submit to Moodle. We will evaluate your output by computing the average log probability of the true values of the unobserved variables.