

MACHINE LEARNING COMS 4771, HOMEWORK 6

Assigned April 21, 2013. Due May 2, 2013 before 1:00pm.

Here are the instructions for submitting your homework. Archive/package all of the files you are submitting as a single tarball or zip archive: “UNI-HW6.tar.gz” or “UNI-HW6.zip”. For example, a compressed tarball would be “ir2322-HW6.tar.gz”. Your homework should contain:

- a writeup (PDF, TXT, or PostScript)
- code (as Matlab M files, shorter code is generally better but include comments)
- any figures/pictures not included in the writeup (PDF or PostScript)
- if you have special instructions, include them as a plain text file called README.txt.

Submit your homework through CourseWorks by doing the following:

- 1 Log into <https://courseworks.columbia.edu/>
- 2 Click “Assignments” on the left side.
- 3 Choose the appropriate HW Folder to submit to.
- 4 Use the filename “yourUNI-HW6.tar.gz” or “yourUNI-HW6.zip”.
- 5 Make sure that the “title” is yourUNI-HW6 (example: zz9999-HW6).
- 6 Add any special instructions in both the description and the README.txt.
- 7 Click “Submit” at the bottom to upload your file.
- 8 If you submit multiple times, only the last submission prior to the deadline will count.
- 9 If something goes wrong, ask the TAs for help.
- 10 In a dire emergency, if nothing else works, send your homework to the TAs.

Handwritten writeups are not allowed without prior approval.

All your code should be written in Matlab (other languages may be used only with prior permission from an instructor). Please submit all your source files, each function in a separate file. Clearly denote what each function does, its inputs and outputs, and to which problem it belongs. Do not resubmit code or data provided to you. Do not submit code written by others. Identical submissions will be detected and both parties will get zero credit. Sample code is available on the Tutorials web page. Datasets are available from the Handouts web page. You may include figures directly in your write-up, or separately and refer to them by filename.

Each homework counts equally towards your grade (other than your worst which will be dropped). Points shown here for each problem indicate relative weights for this specific homework. As always, up to 10% bonus points are available for exceptional, relevant work going beyond what is asked.

1 Conditional independence (15 points)

Let W, X, Y and Z be random variables. Show the following:

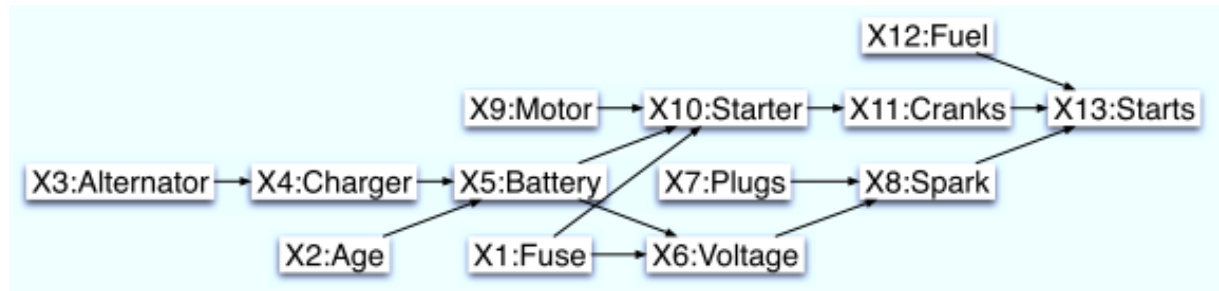
- Prove that $X \perp Y|Z$ iff the joint probability $p(x, y, z)$ can be expressed in the form $a(x, z)b(y, z)$.
- Prove, or disprove with a counter-example, that $X \perp Y|Z$ and $X \perp W|(Y, Z)$ together imply $X \perp (W, Y)|Z$.

- Prove, or disprove with a counter-example, that $X \perp Y|(Z, W)$ and $X \perp Z|(Y, W)$ together imply $X \perp (Y, Z)|W$.
- **Bonus (5 points)** Given a directed acyclic graphical model composed of nodes X_1, X_2, \dots, X_n , prove that the conditional independent statements $X_i \perp X_{\text{non-descendants of } i \setminus \pi_i} | X_{\pi_i}$ imply that the joint distribution can be decomposed as

$$p(X_1, \dots, X_n) = \prod_{i=1}^n p(X_i | X_{\pi_i})$$

2 Constructing a Junction Tree (20 points)

You are given the following directed graphical model or Bayesian network for determining if a car will start or not. Assume all variables in the network are binary. Also, assume there are no observed nodes or evidence. Construct the corresponding junction tree by moralizing, triangulating and then building a junction tree that satisfies the running intersection property (note that the network below is a pruned version of the original car engine network in the literature).



3 Implementing the Junction Tree Algorithm (20 points)

Consider the family of undirected graphical models known as Markov chains, specifically the example below.



For simplicity, assume all variables in the model are binary. The probability distribution implied by the undirected graphical model is $p(x_1, \dots, x_5) = \frac{1}{2} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \psi_{34}(x_3, x_4) \psi_{45}(x_4, x_5)$. Write an implementation of the junction tree algorithm that computes all the pairwise marginals $p(x_i, x_{i+1})$ for such a Markov chain for any number of variables n and any initialization of the clique potential functions. The initial clique potentials will serve as inputs to your junction tree algorithm and should be a cell array of $n - 1$ elements, each of which is a 2×2 matrix of non-negative values as shown in the following Matlab code:

```
n=5;
psis = cell(n-1,1);
```

```

for i=1:(n-1)
    psis{i}=rand(2,2);
end

```

The output of your junction tree algorithm (JTA) should be an identical data structure. It should contain consistent marginals that sum to unity appropriately and agree pairwise. Since the tree is just a chain, there is no need to implement a recursive algorithm (i.e. the Collect and Distribute steps in the Jordan book). Instead, you need only perform left to right message passing and then right to left message passing by using a for loop or standard iteration. In other words, the JTA should process the cliques for $i = 1 : n - 1$ and then for $i = n - 1 : -1 : 1$ to handle all the necessary messages in the JTA.

Test your algorithm by recovering the pairwise marginals when given the following values for the potential functions in the graphical model:

$$\psi_{12}(x_1, x_2) = \begin{matrix} & x_2 = 0 & x_2 = 1 \\ \begin{matrix} x_1 = 0 \\ x_1 = 1 \end{matrix} & \begin{pmatrix} 0.3 & 0.9 \\ 0.8 & 0.2 \end{pmatrix} \end{matrix}$$

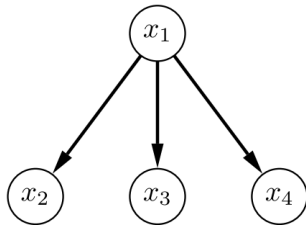
$$\psi_{23}(x_2, x_3) = \begin{matrix} & x_3 = 0 & x_3 = 1 \\ \begin{matrix} x_2 = 0 \\ x_2 = 1 \end{matrix} & \begin{pmatrix} 0.5 & 0.1 \\ 0.1 & 0.5 \end{pmatrix} \end{matrix}$$

$$\psi_{34}(x_3, x_4) = \begin{matrix} & x_4 = 0 & x_4 = 1 \\ \begin{matrix} x_3 = 0 \\ x_3 = 1 \end{matrix} & \begin{pmatrix} 0.1 & 0.6 \\ 1.3 & 0.1 \end{pmatrix} \end{matrix}$$

$$\psi_{45}(x_4, x_5) = \begin{matrix} & x_5 = 0 & x_5 = 1 \\ \begin{matrix} x_4 = 0 \\ x_4 = 1 \end{matrix} & \begin{pmatrix} 0.3 & 0.3 \\ 0.1 & 0.9 \end{pmatrix} \end{matrix}$$

4 ArgMax Junction Tree (20 points)

Consider the Bayesian network below over binary variables x_1, x_2, x_3, x_4 .



You are given the conditional distributions:

$$\begin{aligned}
 p(x_1) &= \frac{x_1 = F}{2/3} \mid \frac{x_1 = T}{1/3} \\
 p(x_2|x_1) &= \frac{x_1 = F}{x_1 = T} \mid \frac{x_2 = F}{1/2} \mid \frac{x_2 = T}{1/2} \\
 &\quad \frac{x_1 = F}{x_1 = T} \mid \frac{x_2 = F}{1/4} \mid \frac{x_2 = T}{3/4} \\
 p(x_3|x_1) &= \frac{x_1 = F}{x_1 = T} \mid \frac{x_3 = F}{1/8} \mid \frac{x_3 = T}{7/8} \\
 &\quad \frac{x_1 = F}{x_1 = T} \mid \frac{x_3 = F}{3/4} \mid \frac{x_3 = T}{1/4} \\
 p(x_4|x_1) &= \frac{x_1 = F}{x_1 = T} \mid \frac{x_4 = F}{1/4} \mid \frac{x_4 = T}{3/4} \\
 &\quad \frac{x_1 = F}{x_1 = T} \mid \frac{x_4 = F}{1/2} \mid \frac{x_4 = T}{1/2}
 \end{aligned}$$

Build a junction tree, keeping the maximum clique size as small as possible. Run the ArgMax junction tree algorithm on it. Show the final states of all potential functions and separators, and use these to find the most likely configuration of x_1, x_2, x_3, x_4 .