CSE 5522 Homework 1

Manjari Akella

9/20/2013

1). Here's a puzzle about cars:

a. You work in the back of an ice cream shop (in the US) that sells chocolate and vanilla ice cream cones. In a forced choice, 70% of men prefer chocolate over vanilla ice cream. 80% of women prefer chocolate over vanilla. 50.8% of the US population is female. An order comes in for a vanilla ice cream cone. What is the probability that the customer is female?

Ans. Let us first define the random variables and the values they can take. The random variables in the system are –
I = Ice cream which can take values <v, c> where v = vanilla, c = chocolate
G = Gender which can take values <m, f> where m = male, f = female

Now, let us write down the probabilities given. We have –
P(I = c | G = m) = 0.7
So, P(I = v | G = m) = 0.3

P(I = c | G = f) = 0.8
So, P(I = v | G = f) = 0.2

P(G = f) = 0.508
So, P(G = m) = 0.492

We have to calculate –
P(G = f | I = v) = ?

Using Bayes' theorem, we have –

$$P(G = f \mid I = v) = \frac{P(I = v \mid G = f) * P(G = f)}{P(I = v)}$$

Using alpha normalization and substituting the values we know,
P(G = f | I = v) = α 0.2 * 0.508
= α 0.1016
To compute α, we have
α = 1 / (P(I = v | G = f) * P(G = f)) + (P(I = v | G = m) * P(G = m))
= 1 / (0.2*0.508) + (0.3*0.492)
= 1 / (0.1016) + (0.3*0.492)
= 1 / (0.1016) + (0.1476)
= 1 / 0.2492

Substituting this we get

$P(G = f \mid I = v) = \alpha\ 0.1016$

$= 0.1016/0.2492 = 0.4077$

Hence if an order comes in for vanilla, the probability that it was given by a female is 0.4077

b. Draw a Bayesian network that describes the situation, including the complete set of Conditional Probability Tables.

Ans.

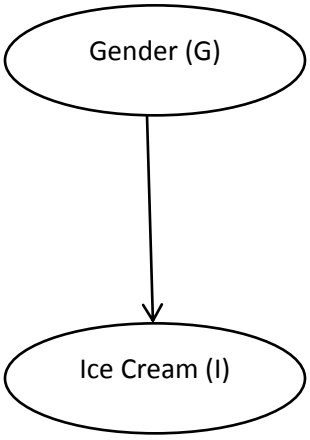


Figure 1: Bayesian Network for the problem

The following is the probability distribution at each node.

At the node 'Gender' –

	P(G)
male	0.492
female	0.508

At the node 'Ice Cream' –

	P(I|G)
male	<0.3,0.7>
female	<0.2,0.8>

2. Let D, E, F, G, and H be five discrete random variables. Assume that I have given you a distribution for P(E), P(D|E), P(F|E), P(G|D,F), P(H|F). Moreover, I will tell you that there are conditional independence assumptions P(F|D,E)=P(F|E), P(G|D,E,F)=P(G|D,F) and P(H|D,E,F,G) = P(H|F).

a. Show, with explicit steps in the derivation, how you could compute P(D|E,F,G,H) in terms of only the given distributions.

Ans. We need to find P(D|E,F,G,H). Using the definition of conditional probability, we have -

$$P(D|E,F,G,H) = \frac{P(D,E,F,G,H)}{P(E,F,G,H)}$$

Let us first solve the numerator. We can define the joint probability as a product of conditional probabilities as follows

P(D,E,F,G,H) = P(H|G,F,E,D)*P(G|F,E,D)*P(F|E,D)*P(D|E)*P(E)
But using the given conditional independence assumptions, we can rewrite this as –
= P(H|F)*P(G|D,F)*P(F|E)*P(D|E)*P(E)

For solving the denominator, we need to sum the joint over the random variable D. So,

$$P(E,F,G,H) = \sum_{d \epsilon D} P(D=d,E,F,G,H)$$
$$= \sum_{d \epsilon D} P(H|F)*P(G|D=d,F)*P(F|E)*P(D=d|E)*P(E)$$
$$= P(H|F)*P(F|E)*P(E) \sum_{d \epsilon D} P(G|D=d,F)*P(D=d|E)$$

Dividing the numerator and denominator, we have

$$P(D|E,F,G,H) = \frac{P(D,E,F,G,H)}{P(E,F,G,H)}$$
$$= \frac{P(H|F)*P(G|D,F)*P(F|E)*P(D|E)*P(E)}{P(H|F)*P(F|E)*P(E) \sum_{d \epsilon D} (G|D=d,F)*P(D=d|E)}$$
$$= \frac{P(G|D,F) * P(D|E)}{\sum_{d \epsilon D} P(G|D=d,F)*P(D=d|E)}$$

b. Draw a Bayesian network for the above distributions. You need not include CPTs since I didn't give any here.

Ans.

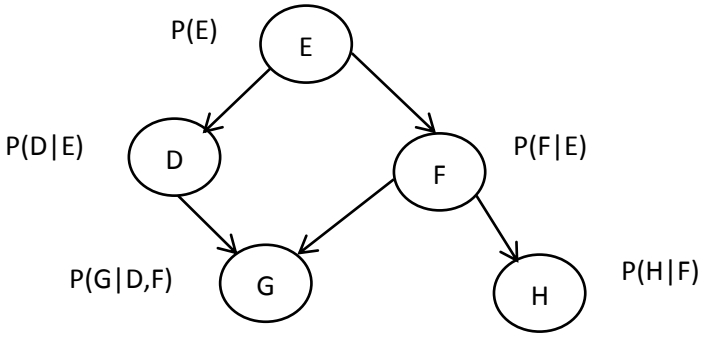
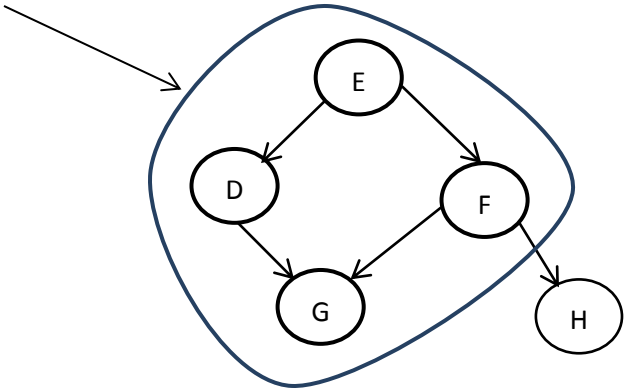


Figure 2: Bayesian Network for the problem

c. Which nodes are in D's Markov blanket?

Ans. Markov's blanket of a node is defined as the set of nodes which constitute the parents, children and children's parents. So for node D we have

Markov's blanket = E,G,F



d. Add a node X to the network such that X is not in D's Markov blanket

---

Ans.

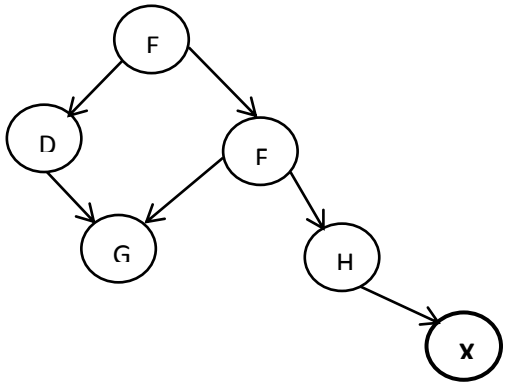


Figure 3: Bayesian Network including node X

3. Polytrees

a. Draw one Bayesian network that is a polytree and one that is not a polytree. (Label clearly.)

Ans.

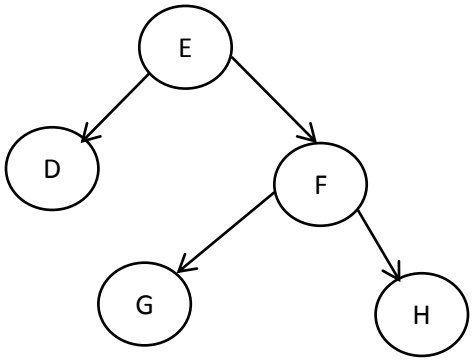


Figure 4: Bayesian Network which is a polytree

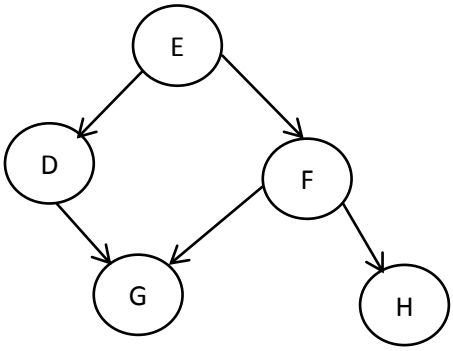
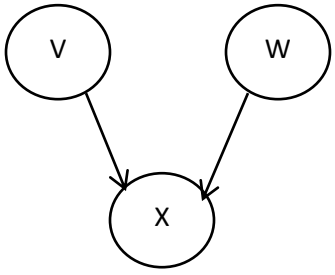
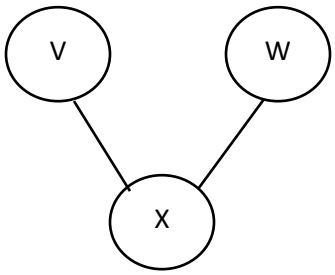


Figure 5: Bayesian Network which is not a polytree

b. In a three-node network, V and W are parents of X. Neither W nor V has a parent. Is this a polytree? Why or why not?

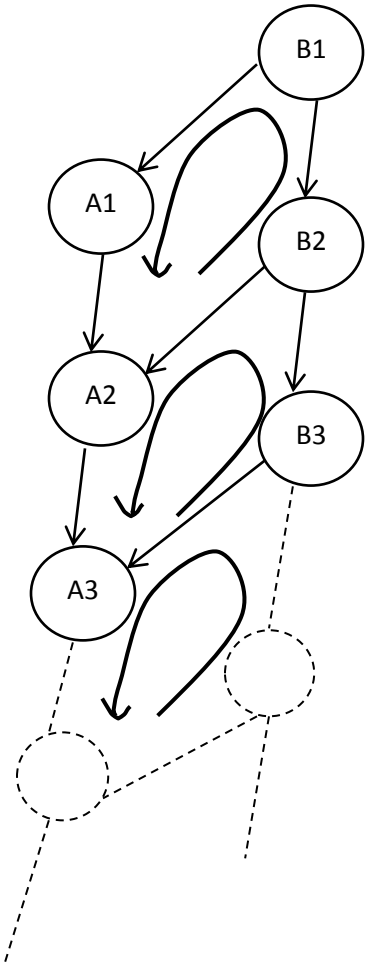
Ans. A polytree is defined as a graph where there is only a single path between any two nodes. In other words, if we replace its arcs with edges, we obtain an undirected graph that is both connected and acyclic. In this structure, we indeed have just a single path between a pair of nodes. Also if we replace arcs with edges, we obtain an undirected graph that is still acyclic.





c. B1 is parent of B2 who is parent of B3 and so forth. A1's parent is B1, A2's parent is B2 and A1, A3's parent is B3 and A2, and so forth. Is this repeating structure a polytree?

Ans.



No, this repeating structure is not a polytree. This is because there exists more than one path between a pair of nodes. Also, if we replace arcs with edges, the structure no longer remains acyclic (the curved arrows in the figure above represent the cycles in such a graph).

d. Explain why knowing if you have a polytree is important for inference.

Ans. Information about whether or not we have a polytree is essential for inference. It gives us an insight into the time taken to reach such an inference. The belief propagation in a polytree structure is easier to keep track of than in a non-polytree. While propagating any changes caused at a node in a polytree, we need to keep a track of a linear number of changes whereas in a non-polytree, the changes we need to keep track of will be exponential. In other words, the time complexity of inference is linear in the size of the network of a polytree. This may, in the worst case, be exponential for a non-polytree. Polytrees enable easier message passing and hence are much more efficient as tools for inference.

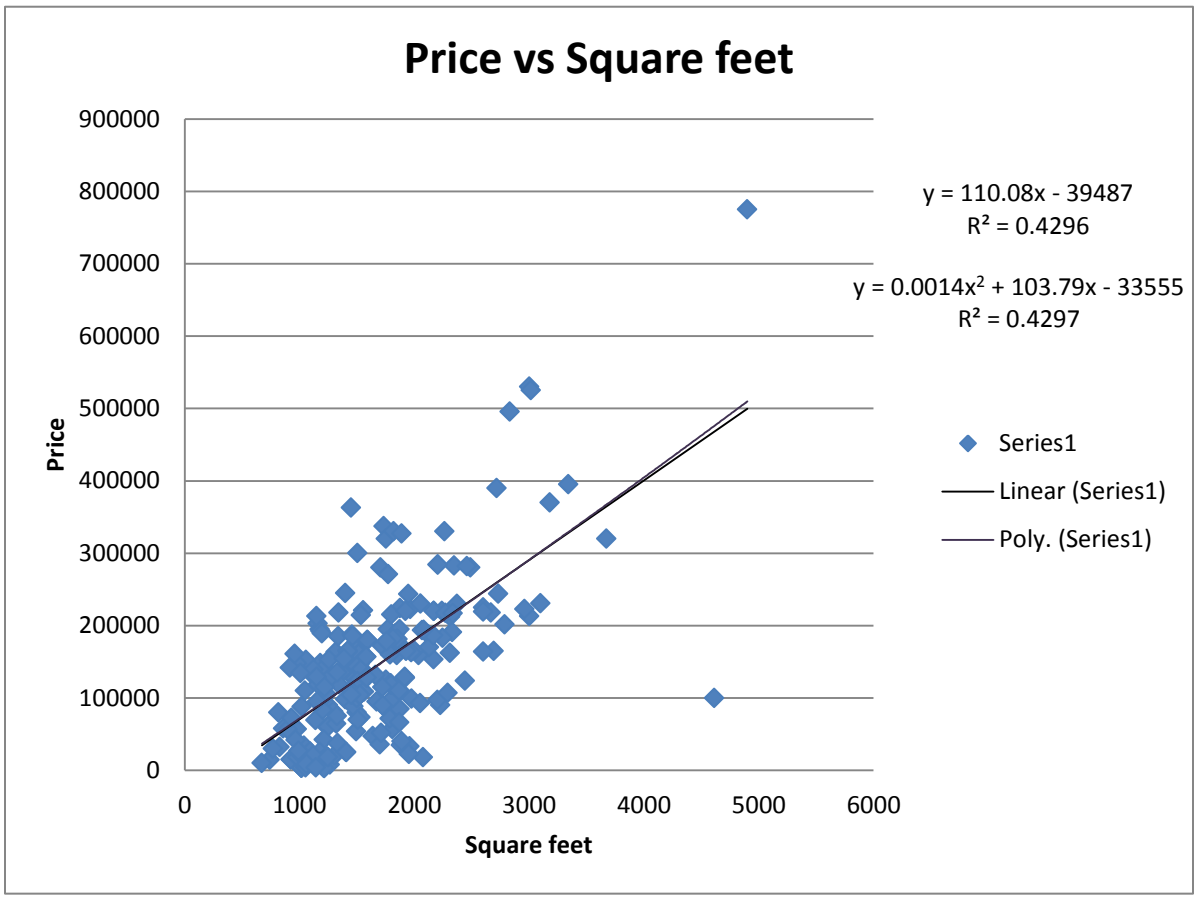
4.(40 points + 15 bonus points) In class, I discussed regression models for predicting the price of a house given it size in square feet. The datafile I used for that class can be found on the Carmen website.

a. (5 points) Write a program to read in the data file and compute the mean and standard deviation, min and max for both price and square feet. Please do not use built-in functions for a language.Note: we will evaluate your code on another dataset, so your program should take the name of an input file as an argument. Also, don't throw away this code, we will use it in a later assignment.

Ans.

b. (20 points) Following equation 18.3 on p719 (and the slides), modify your program to analytically compute the linear regression coefficients w0 and w1 for the input data file.

Ans. The following plot shows a linear and a quadratic curve fitted to the data.



c. (15 points) Following equations 18.4-18.5 (and the discussion following them), write a program to start from a non-optimal w0 and w1 and update using batch gradient descent. Use a very small alpha (for this data set, say alpha=0.0000000001), and start with initial choice of [w0,w1] to be your answer from the last question plus [100,100]. Iterate until convergence (where the difference in the new w0,w1] and old [w0,w1] is less than epsilon, where you define epsilon). How many iterations did it take to converge? Do you get the same answer as in the previous question? What happens if you make alpha orders of magnitude bigger? Smaller? (Explicitly address these questions in your writeup.)

Ans.

- Using the gradient descent algorithm and updating the weights according to the formulas, the value of w1 converges in most of the cases examined.
- The change in w0 for each iteration however, is very small. This makes sense because the scale for this data(price) is different that of the square feet.

- Just using these values as it is, the following were the results. Alpha values larger than 1e-9, lead to both w0 and w1 diverging NaN pretty quickly. This is exactly what we expect. As values of alpha increase, the weights tend to diverge.
- As alpha becomes smaller than 1e-9, w1 converges. The number of iterations needed for this increases with decreasing alpha which is what we expect to happen. At 1e-10, it takes 121 iterations whereas for 1e-11, it takes 924 iterations
- Further, there is a relation between the values of alpha and epsilon. As alpha is reduced, we need to increase epsilon to achieve convergence on w1.
- So, w1 converges to the answer in the previous part (optimal solution) within a slight error.
- w0, however, doesn't converge to the optimal value. In fact, the change in its value after each iteration is so small, we need a very small epsilon to allow it to converge. Further, it is most likely to take a lot of iterations since change in each iteration is very small (in the order of 10^{-6}).
- The following are the results with different alpha and epsilon values –

alpha	epsilon	Iterations	w1 convergence value	w0 convergence value
$1e^{-5}$	0.01	80	NaN	NaN
$1e^{-7}$	0.01	164	NaN	NaN
$1e^{-8}$	0.01	382	NaN	NaN
$1e^{-9}$	0.01	8	110.0335399129	-39386.5576787586
$1e^{-10}$	0.01	90	110.1627087828	-39386.5051300708
$1e^{-10}$	0.001	121	110.0420811154	-39386.5436670659
$1e^{-11}$	0.01	600	111.4337170162	-39386.5051300708
$1e^{-11}$	**0.001**	**924**	**110.1687231308**	**-39386.5051300708**
$1e^{-12}$	0.01	2764	124.1408573658	-39386.5051300708
$1e^{-12}$	0.001	6013	111.4399244036	-39386.5051300708
$1e^{-13}$	0.01	-	210.0807802353	-39386.5051300708
$1e^{-13}$	0.001	27645	124.1446718177	-39386.5051300708
$1e^{-13}$	0.0001	60149	111.4399420751	-39386.5051300708
$1e^{-14}$	0.00001	601507	111.4399638191	-39386.5051300708

- In my opinion, however, using these values as it is, is not a fair thing to do.
- Since both y and x data have different scale, their change after each iteration will be proportional to it.
- This is the reason w1 doesn't converge.
- The solution to this might be using some sort of normalization to bring them to the same scale.

d. (Bonus: 5 points) Implement stochaistic gradient descent, where you randomly select training points and update after each one. (Your alpha parameter will need to be different. Why and how?) Does it converge? If so, does it converge more quickly in terms of number of updates? number of points examined?

Ans. Stochastic gradient descent will lead to lesser iterations as the alpha is increased to converge.

e. (Bonus: 10 points) Implement gradient descent for learning quadratic regression (i.e. y=w2*x^2 + w1*x + w0). You'll need to work out for yourself, or do some research, into what the update equations should be.

Ans. For a quadratic regression, we try to fit a parabola over our data points. So, we have the equation –

$$y=w_2*x^2+w_1*x+w_0$$

Here we have to find the update equations for the weights w_2,w_1,w_0.
We know that for the gradient descent algorithm, the form these update equations take is –

$$w_i = w_i - \alpha \ \partial / \partial \ w_i \ (\text{Loss}(w)) \ldots\ldots\ldots\ldots\ldots\ldots \text{eq. 1}$$

The value of $\partial / \partial \ w_i \ (\text{Loss}(w))$ can be calculated as follows –

$$\partial / \partial \ w_i \ (\text{Loss}(w)) = \partial / \partial \ w_i \ (y-h_w(x))^2$$
$$= 2 \ (y - h_w(x))* \ \partial / \partial \ w_i \ (y-h_w(x))$$
$$= 2 \ (y - h_w(x))* \ \partial / \partial \ w_i \ (y-(w_2*x^2+w_1*x+w_0))$$

Computing the second half of the product for each w_2,w_1,w_0 –

$$\partial / \partial \ w_0 \ (y-(w_2*x^2+w_1*x+w_0)) = -1$$

$$\partial / \partial \ w_1 \ (y-(w_2*x^2+w_1*x+w_0)) = -x$$

$$\partial / \partial \ w_2 \ (y-(w_2*x^2+w_1*x+w_0)) = -x^2$$

So we have

$$\partial / \partial \ w_0(\text{Loss}(w)) = -2 \ (y - h_w(x))$$

$$\partial / \partial \ w_1(\text{Loss}(w)) = -2x \ (y - h_w(x))$$

$$\partial / \partial \ w_2(\text{Loss}(w)) = -2 \ x^2(y - h_w(x))$$

Substituting these values in eq.1, folding the –2 into the value of α we get the following update equations for each of the weights –

$$w_0 = w_0 - \alpha \ \partial / \partial \ w_0(\text{Loss}(w))$$

$= w_0 - \alpha (-2 (y - h_w(x)))$

$= w_0 + \alpha (y - h_w(x))$

$w_1 = w_1 - \alpha \ \partial / \partial \ w_1(\text{Loss}(w))$

$= w_1 - \alpha (-2 x(y - h_w(x)))$

$= w_1 + \alpha x(y - h_w(x))$

$w_2 = w_2 - \alpha \ \partial / \partial \ w_2(\text{Loss}(w))$

$= w_2 - \alpha (-2 (y - h_w(x)))$

$= w_2 + \alpha x^2(y - h_w(x))$

For N training examples, we have summation over it. So that

$w_0 = w_0 + \alpha \sum_j \ (y_j - h_w(x))$

$w_1 = w_1 + \alpha \sum_j \ x_j(y_j - h_w(x))$

$w_2 = w_2 + \alpha \sum_j \ x_j^2(y_j - h_w(x))$

Using these update equations, we can easily modify the program to implement gradient descent for learning quadratic regression.

Further to compute the optimal values of w2,w1,w0, we have the following matrix equation –

$$\begin{bmatrix} \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^3 & \sum x_i^2 & \sum x_i \\ \sum x_i^2 & \sum x_i & N \end{bmatrix} \begin{bmatrix} w2 \\ w1 \\ w0 \end{bmatrix} = \begin{bmatrix} \sum x_i^2 y_i \\ \sum x_i y_i \\ \sum y_i \end{bmatrix}$$