15-381 Spring 06 Assignment 6: Neural Nets, Cross-Validation and Bayes Nets

Questions to Sajid Siddiqi (siddiqi@cs.cmu.edu)

Out: 4/17/06   Due: 5/02/06

Name: ______________________   Andrew ID: ______________________

Please turn in your answers on this assignment (extra copies can be obtained from the class web page). This written portion must be turned in at the beginning of class at 1:30pm on May 2^{nd}. There is no programming assignment in this homework. Please write your name and Andrew ID in the space provided on the first page, and write your Andrew ID in the space provided on each subsequent page. This is worth 5 points: if you do not write your name/Andrew ID in every space provided, you will lose 5 points.

Late policy. Both your written work and code are due at 1:30pm on 3/30. Submitting your work late will affect its score as follows:

- If you submit it after 1:30pm on 5/02 but before 1:30pm on 5/03, it will receive 90% of its score.
- If you submit it after 1:30pm on 5/03 but before 1:30pm on 5/04, it will receive 50% of its score.
- If you submit it after 1:30pm on 5/04, it will receive no score.

Collaboration policy. You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas in the class in order to help each other answer homework questions. You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other the answers
- not to copy answers
- not to allow your answers to be copied

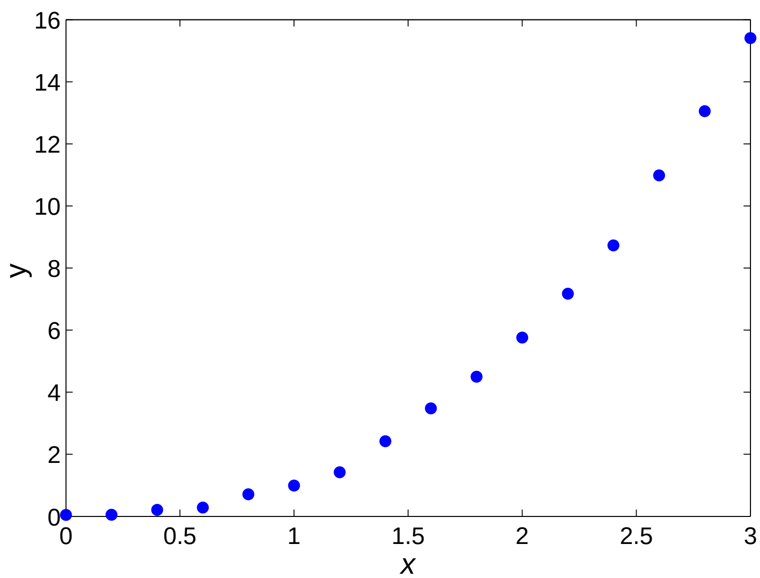
In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we ask that you specifically record on the assignment the names of the people you were in discussion with (or "none" if you did not talk with anyone else). This is worth five points: for each problem, space has been provided for you to either write people's names or "none". If you leave any of these spaces blank, you will lose five points. This will help resolve the situation where a mistake in general discussion led to a replicated weird error among multiple solutions. This policy has been established in order to be fair to the rest of the students in the class. We will have a grading policy of watching for cheating and we will follow up if it is detected. For the programming part, you are supposed to write your own code for submission.

1 Regression and Gradient Descent

References (names of people I talked with regarding this problem or "none"):

____

We will design a gradient descent approach for fitting some data that is given to us. Suppose we have N input-output pairs $\{(x_i, y_i)\}_{i=1}^N$. Our goal is to find the parameters that predict the output y from the input x according to some function $y = f(x)$. The pairs are plotted in the graph below.



1.1 (3 points)

Just by looking at the plot intuitively, which one of the following is the best choice for a function $y = f(x)$ in order to fit the given data? Assume w is some parameter.

1. $y = wx$
2. $y = x^w$
3. $y = \sqrt[w]{x}$

1.2 (3 points)

For any setting of w, we can measure how well a function fits the data. According to your function f above, write down the sum-of-squared-error function E between predictions y and inputs x.

1.3 (5 points)

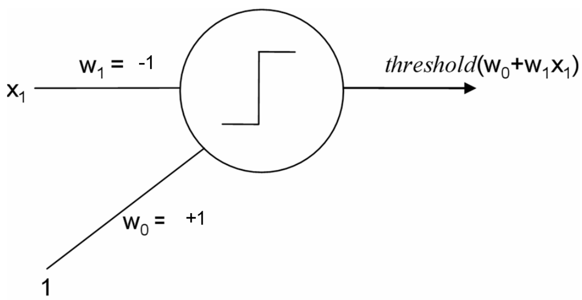
The parameter w can be determined iteratively using gradient descent. For the error function you formulated above, derive the gradient descent update rule $w \leftarrow w - \alpha \frac{dE}{dw}$ and write it down below.

2 Neural Networks

References (names of people I talked with regarding this problem or "none"):

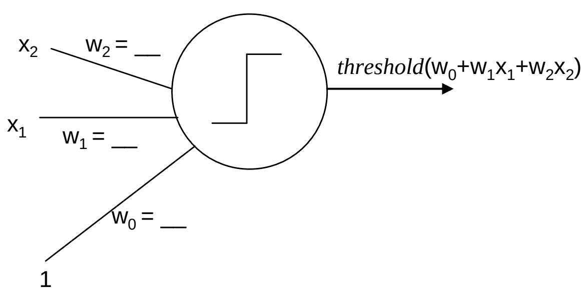
____

We will now build some neural networks to represent basic boolean functions. For simplicity, we use the *threshold* function as our basic units instead of the sigmoid function, where $threshold(t) = +1$ if the input is greater than 0, and 0 otherwise. We have inputs x_i and weights w_i. Suppose we are given boolean input data x_i where +1 represents TRUE and 0 represents FALSE. The boolean NOT function can be represented by a single-layer, single-unit neural net such that the output is +1 if and only if $\neg x_1 = TRUE$:

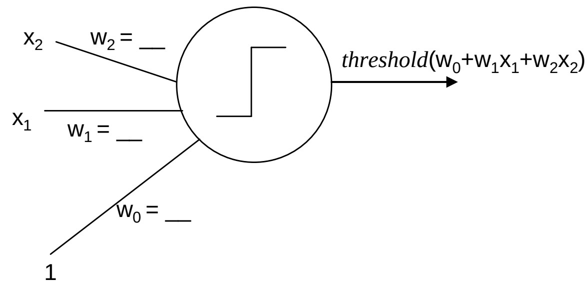


2.1 (10 points)

1. Fill in appropriate weights for the neural net below to represent the OR function:



2. Fill in appropriate weights for the neural net below to represent the AND function:



2.2 (10 points)

Recall that the XOR function for two variables has the following truth table:

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

We saw in class how a single perceptron cannot successfully represent the XOR function. However, a neural net consisting of multiple perceptron units should be able to. Devise a multi-unit neural net (also multi-layer if needed) that computes the XOR of two inputs x_1 and x_2 using the same basic unit as the neural nets above. Draw the layout diagram below and specify the weights of different edges clearly.

3 Cross-Validation

References (names of people I talked with regarding this problem or "none"):

____

We now carry out leave-one-out cross-validation (LOOCV) in a simple classification problem. Consider the following dataset with one real-valued input x and one binary output y. We are going to use k-NN with Euclidean distance to predict y for x.

(a) **(5 points)** What is the LOOCV error of 1-NN on this dataset? Give your answer as the *total number of misclassifications.*

(b) **(5 points)** What is the LOOCV error of 3-NN on this dataset? Give your answer as the *total number of misclassifications.*

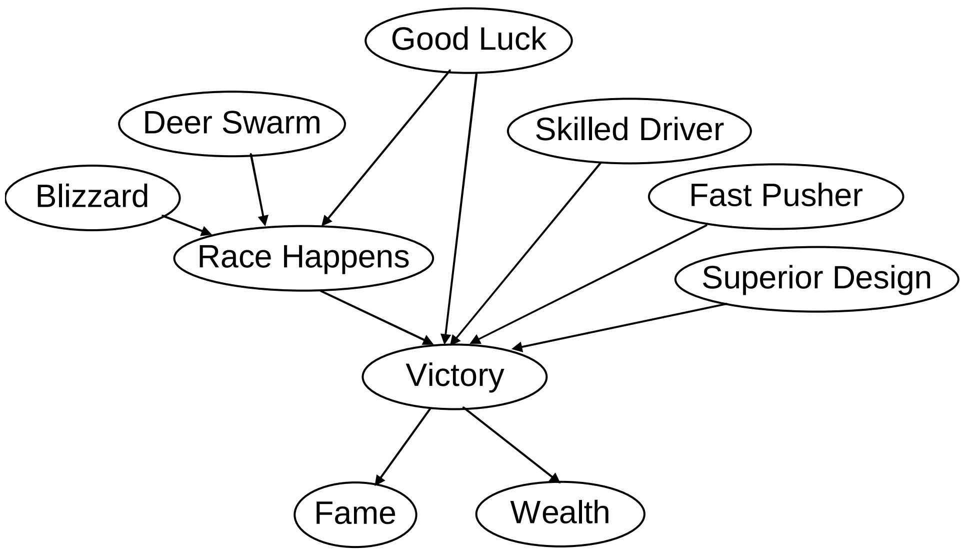
(c) **(3 points)** If you had to pick a classifier for this data, which one of these two would you choose based on their error scores?

4 Conditional Independence in Bayes Nets

References (names of people I talked with regarding this problem or "none"):

____

It's Carnival season, and buggies are all around us. We construct a Bayes Net of boolean variables to represent the probability of our favorite buggy winning the annual buggy race in Schenley Park, the factors involved, and the payoff.



4.1 (18 points)

Use the rules of *d-separation* to say whether the given conditional independencies are implied by the Bayes Net (TRUE/FALSE). The notation $X \perp Z \mid Y$ means that X is conditionally independent of Z given that the value of Y is known.

(a) $SkilledDriver \perp Fame \mid Victory$? ______

(b) $Wealth \perp Blizzard \mid RaceHappens$? ______

(c) $SkilledDriver \perp FastPusher \mid Fame$? ______

(d) $Victory \perp DeerSwarm \mid \{RaceHappens, GoodLuck\}$? ______

(e) $Blizzard \perp DeerSwarm$? ______

(f) $Blizzard \perp DeerSwarm \mid RaceHappens$? ______

4.2 (8 points)

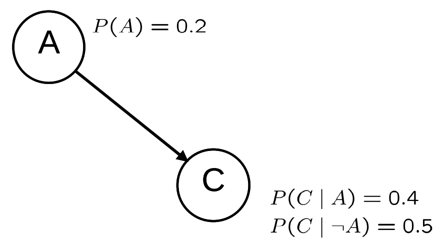
Since this Bayes Net is realistic (in a simplified manner) for this scenario, the dependencies and independencies we get from it should make sense, even if they seem strange at first. For example, *Blizzard* and *DeerSwarm* represent two possible natural disasters that affect the probability of the race occurring or being cancelled (*RaceHappens*). Explain how your answers to (e) and (f) above make sense in the context of this scenario. ***Note:*** *let's assume that Pittsburgh deer don't mind swarming in a blizzard! Otherwise there would be an arrow from Blizzard to DeerSwarm.*

5 Inference in Bayes Nets (20 points)

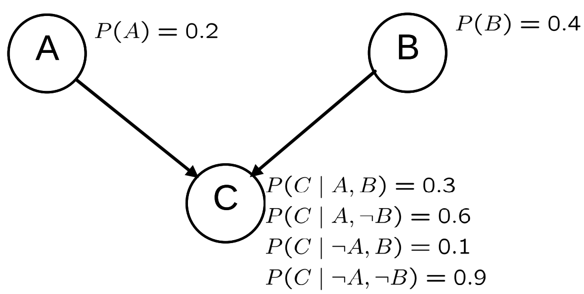
References (names of people I talked with regarding this problem or "none"):

____

Compute the following probabilities from the corresponding Bayes Net. Again, the notation $X \perp Z \mid Y$ implies $P(X \mid Y, Z) = P(X \mid Y)$. Because of independencies implied by the Bayes Net structures (and by using Bayes rule and the chain rule), all the required probabilities are simple to calculate (i.e. 2-3 lines at most). for If you find yourself doing dozens of calculations, stop and look for shortcuts. ***Note:*** *For boolean variables, $P(X)$ is shorthand for $P(X = TRUE)$, and $P(\neg X)$ is shorthand for $P(X = FALSE)$. Similarly, $P(\cdot \mid Y)$ means $P(\cdot \mid Y = TRUE)$.*

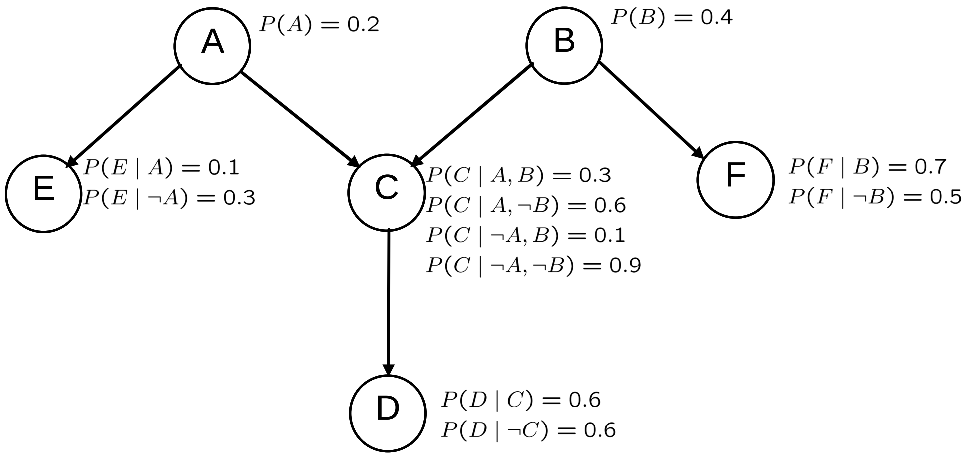


(a) $P(A \mid C)$

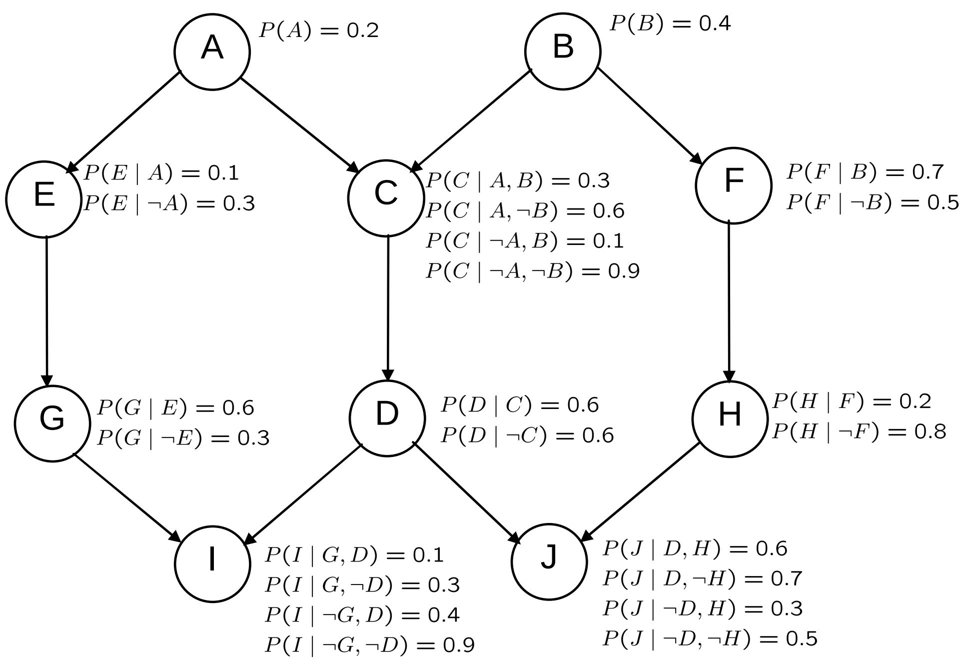


(b) $P(\neg A \mid B)$

(c) $P(\neg A \mid B, \neg C)$



(d) $P(E \mid D)$ (Hint: look at the probabilities. Are all pairs of linked variables actually dependent?)



(e) $P(\neg G \mid \neg J)$ (same hint as above)

6 Bayes Net Structure

References (names of people I talked with regarding this problem or "none"):

____

Consider a robot moving through an environment. At any time, the robot is in some position. While we may not know this position exactly, we might have some beliefs about it. We will model uncertainty in this robot's position with a simple Bayes Net, based on some assumptions we make.

6.1 (5 points)

Assume that the robot is in position X_t at timestep t. Our first simplifying assumption is this: to figure out where the robot might be at time $t+1$, it is sufficient to know the robot's position at time t. Write down a Bayes net structure for timesteps $t = 1, t = 2$ and $t = 3$ consisting of position variables X_t, in a way that reflects this assumption. ***Note:*** *You only have to draw nodes and arrows, not probability tables.*

6.2 (5 points)

Now consider that the robot has a mounted camera (or other sensor), taking an observation of its environment at every timestep. Another sensible assumption is that the observation at time t, depends only upon the position of the robot at that time. Re-draw your Bayes net, this time incorporating the additional assumption above, using the observation variables Y_t along with X_t for all three timesteps. ***Note:*** *You only have to draw nodes and arrows, not probability tables.*

In practice, the true position X_t is usually not known, and the noisy observations Y_t act as 'clues' regarding the true positions. The resulting model is widely used for modeling noisy observations over time and inferring the 'hidden' position or state of a system (e.g. tracking robot position from noisy sensor readings, recognizing words from human utterances).