# MIDTERM: CS 6375
# INSTRUCTOR: VIBHAV GOGATE
# October, 23 2013

**The exam is closed book. You are allowed a one-page cheat sheet. Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, use an additional sheet (available from the instructor) and staple it to your exam.**

- **NAME** _____
- **UTD-ID if known** _____
- SECTION 1: _____
- SECTION 2: _____
- SECTION 3: _____
- SECTION 4: _____
- SECTION 5: _____
- Out of 90: _____

# SECTION 1: SHORT QUESTIONS (15 points)

1. (3 points) The Naive Bayes classifier uses the maximum a posteriori or the MAP decision rule for classification. True or False. Explain.
   **Solution:** True. The decision rule for the Naive Bayes classifier is:

$$\arg; \max_y P(Y = y) \prod_i P(X_i|Y = y)$$

   One can think of $P(Y = y)$ as the prior distribution and $P(X_i|Y = y)$ as the data likelihood.

   Note that when we do the learning, we are using the MLE approach. The decision rule is using MAP inference but the learning algorithm is using the MLE approach. Make sure you understand what this distinction means.

2. (6 points) Let $\theta$ be the probability that "Thumbtack 1" (we will abbreviate it as T1) shows heads and $2\theta$ be the probability that "Thumbtack 2" (we will abbreviate it as T2) shows heads. You are given the following Dataset (6 examples).

| T1 | T2 | T1 | T1 | T2 | T2 |
|---|---|---|---|---|---|
| Tails | Heads | Tails | Tails | Heads | Heads |

   - What is the likelihood of the data given $\theta$.
     **Solution:** We have 3 tails for T1 and 3 heads for T2.
     Therefore, the likelihood is

$$L(\theta) = (1 - \theta)^3(2\theta)^3$$

     The Log-likelihood is:

$$LL(\theta) = 3\log(1 - \theta) + 3\log(2) + 3\log(2\theta)$$

   - What is the maximum likelihood estimate of $\theta$. (Hint: Derivative of $\log(x)$ is $1/x$).
     **Solution:** This can be obtained by differentiating the log-likelihood w.r.t. $\theta$, setting it to zero and solving for the value of $\theta$. The derivative of $LL(\theta)$ is:

$$-3\frac{1}{1 - \theta} + 6\frac{1}{2\theta}$$

     Setting it to zero and solving for $\theta$, we get: $\theta = 1/2$.

3. (2 points) If the data is not linearly separable, then the gradient descent algorithm for training a logistic regression classifier will never converge. True or False. Explain your answer.
**Solution:** False. Our objective function is always concave. The algorithm will always converge under mild assumptions.

4. (2 points) If the data is linearly separable, then the 3-nearest neighbors algorithm will always have 100% accuracy on the training set. True or False. Explain your answer.
**Solution:** False. Here is the counter example.

| X | 0.1 | 0.2 | 1 | 7 |
|---|---|---|---|---|
| Class | + | + | + | − |

Here the data is linearly separable. However, the point $(7, -)$ will be misclassified as $+$ by the 3-nearest neighbors algorithm.

5. (2 points) The decision tree classifier has 100% accuracy on the training set (namely, the data is noise-free). Will logistic regression have the same accuracy (100%) on the training set?
**Solution:** False. Decision tree can learn complex surfaces or Boolean functions (e.g., $XOR$). However, logistic regression can only learn linear classification boundaries (e.g., it cannot represent $XOR$).

## SECTION 2: Decision Trees (20 points)

1. (4 points) Let $\mathbf{x}$ be a vector of $n$ Boolean variables and let $k$ be an integer less than $n$. Let $f_k$ be a target concept which is a disjunction consisting of $k$ literals.

   (Let the $n$ variables be denoted by the set $\{X_1, \ldots, X_n\}$. Examples of $f_2$: $X_1 \vee X_2$, $X_1 \vee \neg X_4$, etc. Examples of $f_3$: $X_1 \vee X_2 \vee X_{10}$, $X_1 \vee \neg X_4 \vee X_7$, etc.)

   State the size of the smallest possible consistent decision tree (namely a decision tree that correctly classifies all possible examples) for $f_k$ in terms of $n$ and $k$ and describe its shape.
   **Solution:** The smallest possible decision tree consistent with $f_k$ contains $2k+1$ nodes with $k$ internal nodes corresponding to the attributes and $k+1$ leaf nodes. Each internal node (except the node at depth $k-1$ which has two leaf nodes) has one leaf node corresponding to an assignment of true to the literal and has one child node corresponding to the false assignment to the particular literal.

   We wish to learn a decision tree to help students pick restaurants using three aspects – the price, the location of the restaurant and the speed of service. The data for training the tree is given below, where the target concept is the column labeled "Like?"

| # | Price | Fast? | On Campus? | Like? |
|---|---|---|---|---|
| 1 | $ | No | No | No |
| 2 | $ | Yes | Yes | No |
| 3 | $$ | No | No | No |
| 4 | $$ | Yes | Yes | Yes |
| 5 | $$ | Yes | No | Yes |
| 6 | $$ | No | Yes | Yes |
| 7 | $$$ | No | No | No |
| 8 | $$$ | Yes | Yes | Yes |

2. (4 points) What is the entropy of the collection of examples with respect to the target label (Like?) ?
   **Solution:** We have 4 $+ve$ and 4 $-ve$ examples. Thus the entropy is 1.

## SECTION 2: Decision Trees (20 points) — Continued

I have copied the dataset from the previous page to this page for convenience.

| # | Price | Fast? | On Campus? | Like? |
|---|---|---|---|---|
| 1 | $ | No | No | No |
| 2 | $ | Yes | Yes | No |
| 3 | $$ | No | No | No |
| 4 | $$ | Yes | Yes | Yes |
| 5 | $$ | Yes | No | Yes |
| 6 | $$ | No | Yes | Yes |
| 7 | $$$ | No | No | No |
| 8 | $$$ | Yes | Yes | Yes |

3. (4 points) Compute the information gain of the attribute Price.
**Solution:**

$$IG = 1 - \frac{2}{8}Entropy(2,0) - \frac{1}{2}Entropy(1,3) - \frac{2}{8}Entropy(1,1)$$

4. (4 points) Compute the information gain of the attribute Fast?
**Solution:**

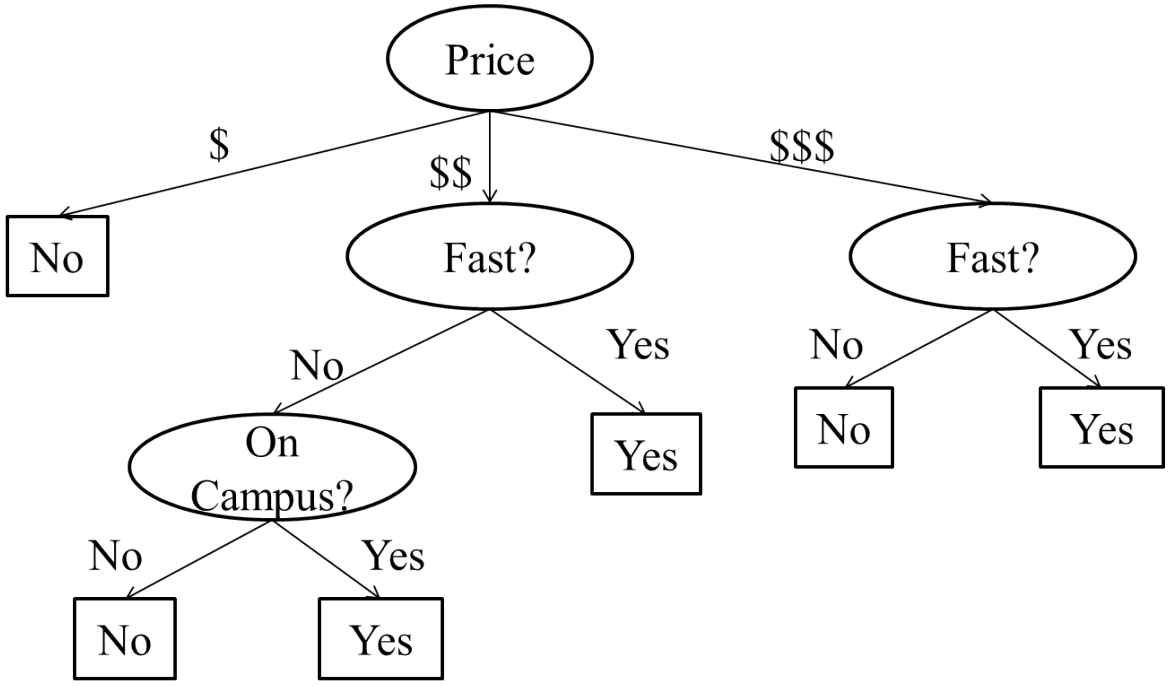
$$IG = 1 - \frac{1}{2}Entropy(3,1) - \frac{1}{2}Entropy(3,1)$$

## SECTION 2: Decision Trees (20 points) — Continued

I have copied the dataset from the previous page to this page for convenience.

| # | Price | Fast? | On Campus? | Like? |
|---|---|---|---|---|
| 1 | $ | No | No | No |
| 2 | $ | Yes | Yes | No |
| 3 | $$ | No | No | No |
| 4 | $$ | Yes | Yes | Yes |
| 5 | $$ | Yes | No | Yes |
| 6 | $$ | No | Yes | Yes |
| 7 | $$$ | No | No | No |
| 8 | $$$ | Yes | Yes | Yes |

5. (4 points) Choose Price as the root node for the decision tree. With this root node, write down a decision tree that is consistent with the data (namely a decision tree that correctly classifies all the training examples). You do not need to learn the decision tree; just make sure it is consistent with the data.
**Solution:**




Note. This is just one of the many possible trees.

## SECTION 3: GRADIENT DESCENT (20 points)

Suppose that the CEO of some startup company calls you and tells you that he sells $d$ items on his web site. He has access to ratings given by $n$ users, each of whom have rated a subset of the the $d$ items. Assume that each rating is a real-number. Your task is to estimate the missing ratings. (Think of the data as $n$ by $d$ matrix in which some entries are missing and your task is to estimate the missing entries).

Being a good machine learner, you have come up with the following model:

$$x_{i,j} = a_i + b_j + u_i v_j$$

where

- $x_{i,j}$ is the rating of the $j$-th product by user $i$
- $a_i$ is the base rating for the user
- $b_j$ is the base rating for each item
- $u_i$ and $v_j$ is some co-efficient for each user and each dimension respectively

Notice that your model has $2(n + d)$ parameters.

1. (5 points) Set up the machine learning problem of estimating the $2(n + d)$ parameters as an optimization task of minimizing the squared error. Write down the formula for the squared error and the optimization task.
   **Solution:** Let $K$ denote the set of entries for which the user ratings are known. The formula for the squared error is:

   $$J(K) = \frac{1}{2} \sum_{x_{i,j} \in K} (x_{i,j} - a_i - b_j - u_i v_j)^2$$

   The optimization problem is:

   $$\arg \min_{a_i, b_j, u_i, v_j} J(K)$$

2. (5 points) Compute the gradient of the error function w.r.t. $a_i$, $b_j$, $u_i$ and $v_j$.
   **Solution:** Skipped: Trivial.

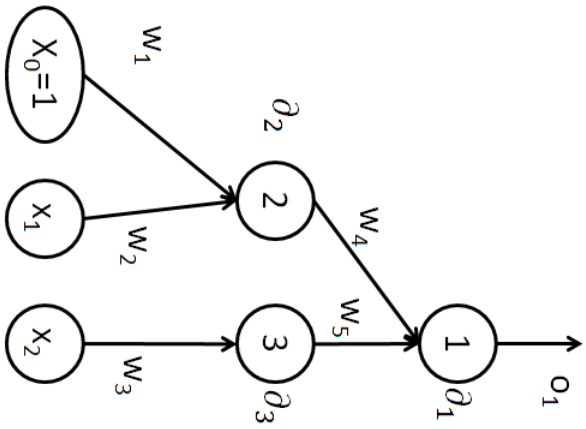
## SECTION 3: GRADIENT DESCENT (20 points) – CONTINUED

3. (5 points) Give the pseudo-code for Batch Gradient Descent for this problem.
**Solution:** Skipped: Trivial.

4. (5 points) Will you use batch gradient descent or incremental (or stochastic or online) gradient descent for this task. How will the data size measured in terms of $n$, $d$ and number of missing entries (let us denote these by $m$) affect your choice?
**Solution:** If $(nd - m)$ is large, we will use the incremental approach. Otherwise, we will prefer the batch approach.

# SECTION 4: NEURAL NETWORKS AND PERCEPTRONS (25 POINTS)

Consider the neural network given below:




Assume that all internal nodes compute the sigmoid function. Write an explicit expression to how back propagation (applied to minimize the least squares error function) changes the values of $w_1$, $w_2$, $w_3$, $w_4$ and $w_5$ when the algorithm is given the example $x_1 = 0$, $x_2 = 1$, with the desired response $y = 0$ (notice that $x_0 = 1$ is the bias term). Assume that the learning rate is $\alpha$ and that the current values of the weights are: $w_1 = 3$, $w_2 = 2$, $w_3 = 2$, $w_4 = 3$ and $w_5 = 2$. Let $o_j$ be the output of the hidden units and output units indexed by $j$.

1. (3 points) Forward propagation. Write equations for $o_1$, $o_2$ and $o_3$ in terms of the given weights and example.

   **Answer:**

   $o_3 = \sigma(w_1 + w_2 x_1) = \sigma(3 + 2 * 0) = \sigma(3)$

   $o_2 = \sigma(w_3 x_3) = \sigma(2 * 1) = \sigma(2)$

   $o_1 = \sigma(w_4 o_2 + w_5 o_3) = \sigma(3\sigma(3) + 2\sigma(2))$

2. (6 points) Backward propagation. Write equations for $\delta_1$, $\delta_2$ and $\delta_3$ in terms of the given weights and example.

   **Answer:**

   $\delta_1 = o_1(1 - o_1)(0 - o_1) = -o_1^2(1 - o_1)$

   $\delta_2 = o_2(1 - o_2)(w_4 \delta_1) = 3o_2(1 - o_2)\delta_1$

   $\delta_3 = o_3(1 - o_3)(w_5 \delta_1) = 2o_3(1 - o_3)\delta_1$

3. (5 points) Give an explicit expression for the new weights.

   **Answer:**

   $w_1 = w_1 + \eta \delta_2 x_0 = 3 + \eta \delta_2$

   $w_2 = w_2 + \eta \delta_2 x_1 = 2$

$w_3 = w_3 + \eta\delta_3 x_2 = 2 + \eta\delta_3$

$w_4 = w_4 + \eta\delta_1 o_2 = 3 + \eta\delta_1 o_2$

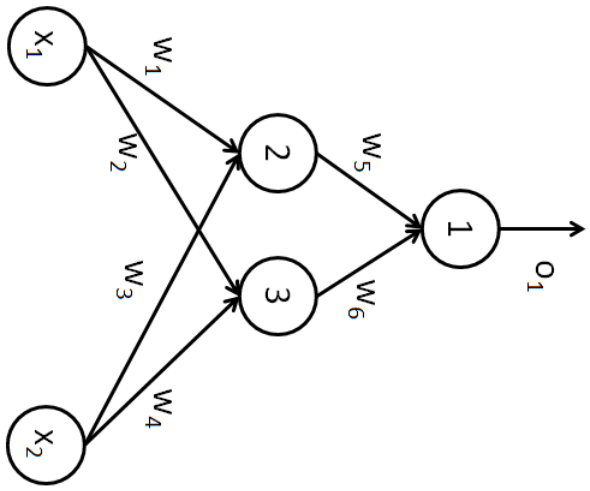
$w_5 = w_5 + \eta\delta_1 o_3 = 2 + \eta\delta_1 o_3$

4. (5 points) Draw a neural network that represents the following Boolean function: $(X_1 \wedge X_2) \vee (\neg X_1 \wedge \neg X_2)$

   **Answer:** Trivial: Skipped.

## SECTION 4: NEURAL NETWORKS AND PERCEPTRONS (25 POINTS) – Continued

Consider the neural network given below. Assume that each unit $j$ is a linear unit and its output is of the form $o_j = C\sum_{i=1}^{n} w_i x_i$ where $x_i$'s are the inputs to the unit and $w_i$ is the weight on the corresponding edge. For example: $o_2 = C(w_1x_1 + w_3x_2)$.




5. (6 points) Can any function that is represented by the above network also be represented by a Perceptron having a single linear unit of the form $o = P\sum_i w_i x_i$ where $x_i$ are the inputs, $w_i$ are the weights attached to the edges and $P$ is a constant. If your answer is yes, then draw the Perceptron detailing the weights and the value for $P$ (i.e., express the weights, lets say $v_1$ and $v_2$ as well as $P$ of the Perceptron in terms of $w_1$,$w_2$,$w_3$,$w_4$, $w_5$, $w_6$, and $C$.) If your answer is no, explain why not?

**Solution:** Yes. It can be represented using a Perceptron with just two inputs $X_1$ and $X_2$.

$$o_1 = C(w_5C(w_1X_1 + w_3X_2) + w_6C(w_2X_1 + w_4X_3))$$

$$o_1 = C^2(X_1(w_5w_1 + w_6w_2) + X_2(w_5w_3 + w_6w_4)$$

Thus assuming that the new Perceptron output is of the form: $P(v_1X_1 + v_2X_2)$, we have

$$P = C^2$$

$$v_1 = (w_5w_1 + w_6w_2)$$

$$v_2 = (w_5w_3 + w_6w_4)$$

## SECTION 5: K nearest neighbors (10 points)

Let us employ the **leave one out cross validation** approach to choose $k$ in $k$ nearest neighbors. As the name suggests, it involves using a single example from the training data as the test (or validation) data, and the remaining examples as the training data. This is repeated such that each example in the training data is used once as the test data.

Formally, given a classifier $C$, the approach computes the $Error$ of $C$ as follows. Let $d$ be the number of training examples and let us assume that the examples are indexed from 1 to $d$.

- $Error = 0$
- **For** $i = 1$ to $d$
  - Remove the $i$-th example from the training set and use it as your test set
  - Train the classifier $C$ on the new training set
  - If the test example is incorrectly classified by $C$ then $Error = Error + 1$
  - Put back the $i$-th example in the training set
- **Return** $Error$

To select the best classifier among a group of classifiers, you apply the leave-one-out cross validation approach to all the classifiers and choose the one that has the smallest error.

**Dataset**:

| X | -0.1 | 0.7 | 1.0 | 1.6 | 2.0 | 2.5 | 3.2 | 3.5 | 4.1 | 4.9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Class | - | + | + | - | + | + | - | - | + | + |

1. (4 points) What is the leave-one-out cross validation error of 1 nearest neighbor algorithm on the dataset given above (use Euclidean distance). Explain your answer.
   **Solution:** 4.

2. (4 points) What is the leave-one-out cross validation error of 3 nearest neighbor algorithms on the dataset given above (use Euclidean distance). Explain your answer.
   **Solution:** 8.

3. (2 points) Will you choose $k = 1$ or $k = 3$ for this dataset?
   **Solution:** Choose $k = 1$. Because the leave one out cross validation error of $k = 1$ is smaller than that of $k = 3$