Introduction to Artificial Intelligence
CAP 4601
Summer 2013
Midterm Exam

1. Terminology (7 Points).

Given the following task environments, enter their properties/characteristics. The properties/characteristics of the Checkers task environment is provided as an example.

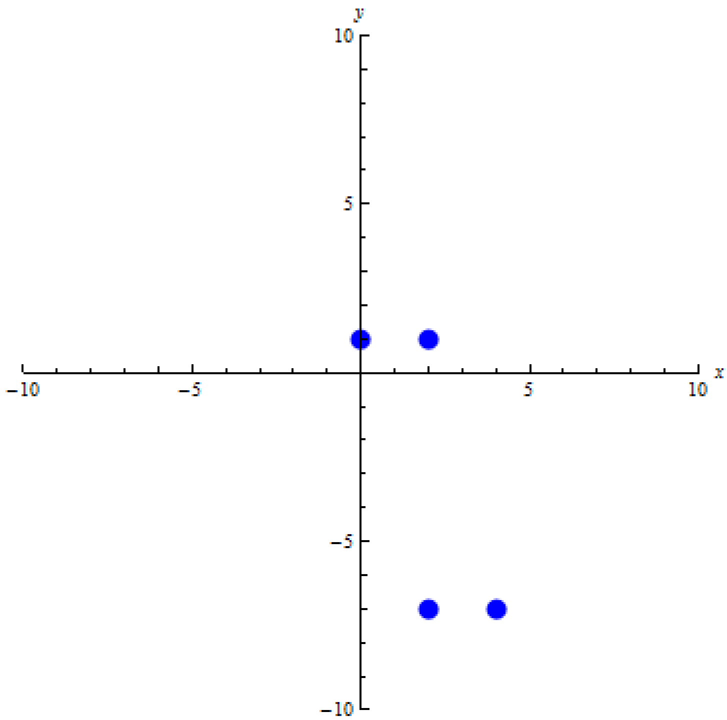
- Task Environment: Checkers
- Fully Observable or Partially Observable: Fully Observable
- Single Agent or Multiagent: Multiagent
- Deterministic or Stochastic: Deterministic
- Episodic or Sequential: Sequential
- Static or Dynamic: Static
- Discrete or Continuous: Discrete
- Known or Unknown: Known

- Task Environment: Robotic Car (Machine Driving)
- Fully Observable or Partially Observable: **Partially Observable**
- Single Agent or Multiagent: **Multiagent**
- Deterministic or Stochastic: **Stochastic**
- Episodic or Sequential: **Sequential**
- Static or Dynamic: **Dynamic**
- Discrete or Continuous: **Continuous**
- Known or Unknown: **Known**

2. Linear Regression (12 Points).

Given the following points: { (0, 1), (2, 1), (2, -7), (4, -7) }

We have the following plot:



Calculate the Sample Mean for the x's (a number): **2**

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{(0)+(2)+(2)+(4)}{4} = \frac{8}{4} = \boxed{2}$$

Calculate the Sample Mean for the y's (a number): **-3**

$$\begin{aligned}\bar{y} &= \frac{\sum_{i=1}^{n} y_i}{n} \\ &= \frac{(1)+(1)+(-7)+(-7)}{4} \\ &= \frac{-12}{4} \\ &= \boxed{-3}\end{aligned}$$

Calculate the Biased Sample Variance for the x's (a number): **2**

$$\begin{aligned}\text{var}_{x,\text{biased}} &= \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2 \\ &= \frac{1}{4}\left(((0)-(2))^2+((2)-(2))^2+((2)-(2))^2+((4)-(2))^2\right) \\ &= \frac{1}{4}\left((-2)^2+(0)^2+(0)^2+(2)^2\right) \\ &= \frac{1}{4}(4+0+0+4) \\ &= \frac{8}{4} \\ &= \boxed{2}\end{aligned}$$

Calculate the Biased Sample Variance for the y's (a number): **16**

$$\begin{aligned}\text{var}_{y,\text{biased}} &= \frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{y})^2 \\ &= \frac{1}{4}\left(((1)-(-3))^2+((1)-(-3))^2+((-7)-(-3))^2+((-7)-(-3))^2\right) \\ &= \frac{1}{4}\left((4)^2+(4)^2+(-4)^2+(-4)^2\right) \\ &= \frac{1}{4}(16+16+16+16) \\ &= \boxed{16}\end{aligned}$$

Calculate the Biased Sample Covariance between the x's and the y's (a number): **-4**

$$
\begin{aligned}
\text{cov}_{(x,y),\text{biased}} &= \frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y}) \\
&= \frac{1}{4}\begin{pmatrix} ((0)-(2))((1)-(-3))+((2)-(2))((1)-(-3))+ \\ ((2)-(2))((-7)-(-3))+((4)-(2))((-7)-(-3)) \end{pmatrix} \\
&= \frac{1}{4}\begin{pmatrix} (-2)(4)+(0)(4)+ \\ (0)(-4)+(2)(-4) \end{pmatrix} \\
&= \frac{1}{4}((-8)+(0)+(0)+(-8)) \\
&= \frac{-16}{4} \\
&= \boxed{-4}
\end{aligned}
$$

Calculate the w_0 from $y = w_0 + w_1*x$ (i.e. calculate the Intercept, a number): **1**

$$
\begin{aligned}
w_1 &= \frac{\text{cov}_{(x,y),\text{biased}}}{\text{var}_{x,\text{biased}}} \\
&= \frac{-4}{2} \\
&= -2 \\
\\
w_0 &= -w_1\bar{x}+\bar{y} \\
&= -(-2)(2)+(-3) \\
&= -(-4)+(-3) \\
&= 4+(-3) \\
&= \boxed{1}
\end{aligned}
$$

Calculate the w_1 from $y = w_0 + w_1*x$ (i.e. calculate the Slope, a number): **-2**

$$
\begin{aligned}
w_1 &= \frac{\text{cov}_{(x,y),\text{biased}}}{\text{var}_{x,\text{biased}}} \\
&= \frac{-4}{2} \\
&= \boxed{-2}
\end{aligned}
$$

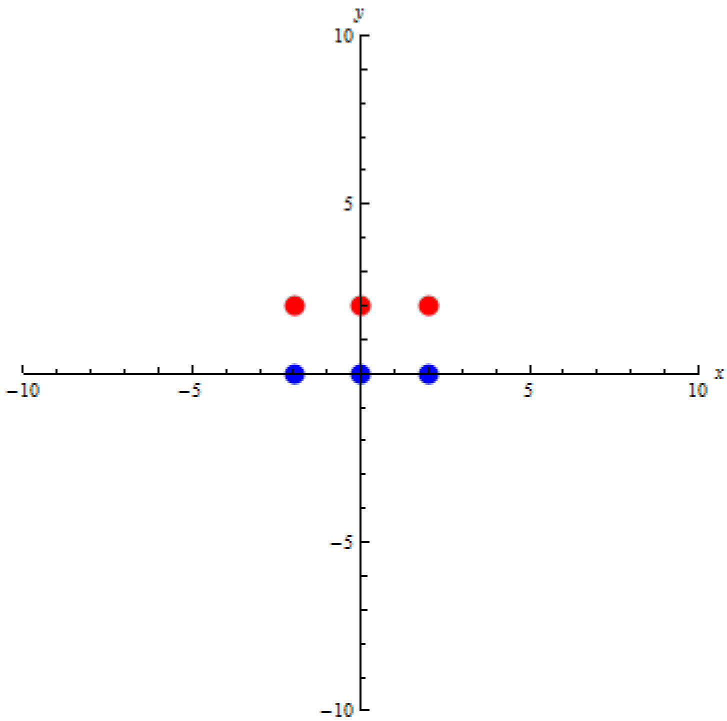
3. Classification (14 Points).

Given the following two classes each with three points, find the mean of each class and then find the equation of the plane that separates those classes halfway between their respective means. Ensure that the separating plane produces positive values for Class 2 Points (Blue) and negative values for Class 1 Points (Red). In other words, find the separating plane that could be used for classification with the maximum margin between the classes and where the separating plane z is z > 0 for Class 2 Points (Blue) and z < 0 for Class 1 Points (Red).

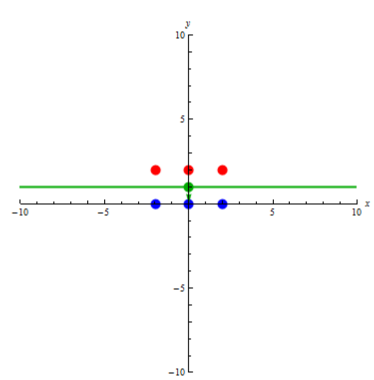
Class 1 Points (Red): { (-2, 2), (0, 2), (2, 2) }

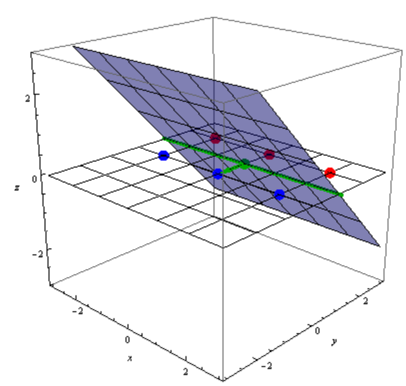
Class 2 Points (Blue): { (-2, 0), (0, 0), (2, 0) }

The following is the plot of these two classes:



We want the following classification:





$$
\begin{aligned}
(\bar{x},\bar{y})_{Class1} &= \left(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n} \right)_{Class1} \\
&= \left(\frac{(-2)+(0)+(2)}{3}, \frac{(2)+(2)+(2)}{3} \right) \\
&= \left(\frac{0}{3}, \frac{6}{3} \right) \\
&= (0,2)
\end{aligned}
$$

$$
\begin{aligned}
(\bar{x},\bar{y})_{Class2} &= \left(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n} \right)_{Class2} \\
&= \left(\frac{(-2)+(0)+(2)}{3}, \frac{(0)+(0)+(0)}{3} \right) \\
&= \left(\frac{0}{3}, \frac{0}{3} \right) \\
&= (0,0)
\end{aligned}
$$

$$
\begin{aligned}
(x,y)_{midpoint} &= \frac{(\bar{x},\bar{y})_{Class1} + (\bar{x},\bar{y})_{Class2}}{2} \\
&= \frac{(0,2)+(0,0)}{2} \\
&= \frac{(0+0,2+0)}{2} \\
&= \frac{(0,2)}{2} \\
&= \left(\frac{0}{2}, \frac{2}{2} \right) \\
&= \boxed{(0,1)}
\end{aligned}
$$

Calculate the x value of the Midpoint between the Sample Mean of the Class 1 Points and the Sample Mean of the Class 2 Points (i.e. the value of x in the midpoint (x, y)): **0**

Calculate the y value of the Midpoint between the Sample Mean of the Class 1 Points and the Sample Mean of the Class 2 Points (i.e. the value of y in the midpoint (x, y)): **1**

Since we want the separating plane to produces positive values for Class 2 Points (Blue), then we want the normalized normal vector to point toward Class 2 Points (Blue). Since we want to find the separating plane used for classification with the maximum margin between the classes, then we want to start point from the midpoint we just calculated; hence, the angle we need to use for this normalized normal vector is $-90^\circ = -\frac{\pi}{2}$. Therefore, we have:

$$\begin{aligned}\vec{\mathbf{n}}_{\text{normalized}} &= \left(\cos(\theta), \sin(\theta)\right)\\ &= \left(\cos(-90^\circ), \sin(-90^\circ)\right)\\ &= \boxed{(0,-1)}\end{aligned}$$

Calculate the x value of the Normalized Normal Vector that points towards the Class 2 Points (Blue) (i.e. the value of the x in the normalized normal vector $\mathbf{n}_{\text{normalized}}$): **0**

Calculate the y value of the Normalized Normal Vector that points towards the Class 2 Points (Blue) (i.e. the value of the y in the normalized normal vector $\mathbf{n}_{\text{normalized}}$): **-1**

Since we have the following:

$$\begin{aligned}f_{classification}(x,y) &= \left(\cos(\theta), \sin(\theta)\right) \bullet \left((x,y) - (x_0, y_0)\right)\\ &= (0,-1) \bullet \left((x,y) - (0,1)\right)\\ &= (0,-1) \bullet (x-0, y-1)\\ &= (0)(x-0) + (-1)(y-1)\\ &= (0)(x) - (0)(0) + (-1)(y) - (-1)(1)\\ &= \left(-(0)(0) - (-1)(1)\right) + (0)(x) + (-1)(y)\\ &= (1) + (0)x + (-1)y\\ &= \boxed{1} + \boxed{0}x + \boxed{-1}y\end{aligned}$$

Calculate the w_0 (a number) in the equation of the separating plane z = $w_0 + w_1*x + w_2*y$ that creates the separating line in the level set z = 0: **1**

Calculate the w_1 (a number) in the equation of the separating plane z = $w_0 + w_1*x + w_2*y$ that creates the separating line in the level set z = 0: **0**

Calculate the w_2 (a number) in the equation of the separating plane z = $w_0 + w_1*x + w_2*y$ that creates the separating line in the level set z = 0: **-1**

4. Naive Bayes Network (24 Points).

Using a small subset of data from an experiment at the Ft. Benning Battle Lab, construct a Naïve Bayes Network to predict a subject's first action (move or fire) in close range engagements under various circumstances.

"**Urban**" means the subject was in a city, vice being in a forest.

"**Day**" means the encounter happened in daylight, vice at night using night-vision equipment.

"**Within25m**" means that the target was first observed at a range less than 25 meters, vice up to 50 meters.

"**MoveFirst**" means that the subject's first action on detecting the threat was to move to cover, vice firing immediately.

HINT: Draw the graph of this Naïve Bayes Network. Remembering that a Naïve Bayes Network assumes conditional independence.

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

QUESTION: Using the data table above, do the following (Do not write any decimal approximations):

1) Write the exact fraction for P(MoveFirst = true). **3/10**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

2) Write the exact fraction for P(MoveFirst = false). **7/10**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

3) Write the exact fraction for P(Urban = false | MoveFirst = false). **4/7**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

4) Write the exact fraction for P(Urban = false | MoveFirst = true). **1/3**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

5) Write the exact fraction for P(Urban = true | MoveFirst = false). **3/7**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

6) Write the exact fraction for P(Urban = true | MoveFirst = true). **2/3**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

7) Write the exact fraction for P(Day = false | MoveFirst = false). **3/7**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

8) Write the exact fraction for P(Day = false | MoveFirst = true). **2/3**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

9) Write the exact fraction for P(Day = true | MoveFirst = false). **4/7**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

10) Write the exact fraction for P(Day = true | MoveFirst = true). **1/3**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

11) Write the exact fraction for P(Within25m = false | MoveFirst = false). **3/7**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

12) Write the exact fraction for P(Within25m = false | MoveFirst = true). **2/3**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

13) Write the exact fraction for P(Within25m = true | MoveFirst = false). **4/7**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

14) Write the exact fraction for P(Within25m = true | MoveFirst = true). **1/3**

Urban	Day	Within25m	MoveFirst
true	true	true	false
true	true	true	false
true	true	true	true
false	true	false	false
true	false	false	false
true	false	false	true
false	false	true	false
false	true	false	false
false	false	false	true
false	false	true	false

15) Write the equation to predict the probability that a subject's first action is to shoot when he encounters a threat within 25 meters in a city at night. In order words, write the equation that could calculate:

P(MoveFirst = false | Urban = true, Day = false, Within25m = true)

in terms of only the probabilities above. Just write the probabilities (using the rules of Probability that you learned about in Chapter 13). Do NOT write the exact fractions nor the decimal approximation.

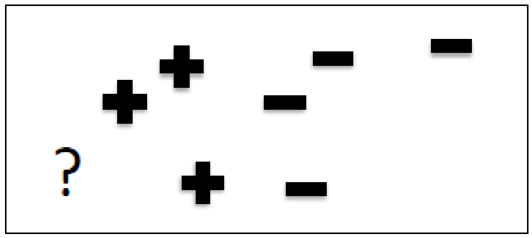
P(MoveFirst = false | Urban = true, Day = false, Within25m = true) = (P(Urban = true | MoveFirst = false) * P(Day = false | MoveFirst = false) * P(Within25m = true | MoveFirst = false) * P(MoveFirst = false)) / (P(Urban = true | MoveFirst = false) * P(Day = false | MoveFirst = false) * P(Within25m = true | MoveFirst = false) * P(MoveFirst = false) + P(Urban = true | MoveFirst = true) * P(Day = false | MoveFirst = true) * P(Within25m = true | MoveFirst = true) * P(MoveFirst = true))

Or written out so that we can actually have a hope of being able to read it:

$$P\left(MoveFirst = false \middle| Urban = true, Day = false, Within25m = true\right) = \frac{P\left(Urban = true \middle| MoveFirst = false\right) P\left(Day = false \middle| MoveFirst = false\right) P\left(Within25m = true \middle| MoveFirst = false\right) P\left(MoveFirst = false\right)}{\left(\begin{array}{l} P\left(Urban = true \middle| MoveFirst = false\right) P\left(Day = false \middle| MoveFirst = false\right) P\left(Within25m = true \middle| MoveFirst = false\right) P\left(MoveFirst = false\right) \\ + \\ P\left(Urban = true \middle| MoveFirst = true\right) P\left(Day = false \middle| MoveFirst = true\right) P\left(Within25m = true \middle| MoveFirst = true\right) P\left(MoveFirst = true\right) \end{array}\right)}$$

5. k-Nearest Neighbor (2 Points).

Given the following labeled data set



For what (minimal) value of k will the query point "?" be negative? Enter "0" if this is impossible. Ties are broken at random – try to avoid them.

7

If $k=1$, then our closest neighbors would be: $\{+\}$ and majority rule would classify our query point as $+$.
If $k=2$, then our closest neighbors would be: $\{+,+\}$ and majority rule would classify our query point as $+$.
If $k=3$, then our closest neighbors would be: $\{+,+,+\}$ and majority rule would classify our query point as $+$.
If $k=4$, then our closest neighbors would be: $\{+,+,+,-\}$ and majority rule would classify our query point as $+$.
If $k=5$, then our closest neighbors would be: $\{+,+,+,-,-\}$ and majority rule would classify our query point as $+$.
If $k=6$, then our closest neighbors would be: $\{+,+,+,-,-,-\}$ and our query point would be arbitrarily classified.
If $k=7$, then our closest neighbors would be: $\{+,+,+,-,-,-,-\}$ and majority rule would classify our query point as $-$.

6. Linear Regression vs. Logistic Regression (6 Points).

Describe the difference between Linear Regression and Logistic Regression. Detail what is being minimized in both methods (i.e. what is being regressed).

Linear Regression is a Supervised Learning method used for Regression, not Classification. Linear Regression minimizes the distance of the y value of each point with the y value of the line of regression.

Logistic Regression is a Supervised Learning method used for Classification, not Regression. Logistic Regression minimizes the distance between the class of a point (expressed as either a 0 or a 1) and the value of that point on the Logistic Function.

7. Parametric vs. Nonparametric Methods (8 Points).

For the methods below, indicate whether the method is parametric or nonparametric using a "P" for parametric and a "N" for nonparametric:

- Linear Regression: P
- k-Nearest Neighbor: N
- Bayesian Network: **P**
- Multivariate Linear Regression: **P**
- Logistic Regression: **P**
- Perceptron: **P**
- Multilayer Feed-Forward Neural Network: **P**
- Locally Weighted Regression: **N**
- Support Vector Machines: **P**
- Kernel Density Estimation: **N**

8. Parametric vs. Nonparametric (4 Points).

Describe the difference between parametric methods and nonparametric methods.

Parametric methods summarize data into a fixed number of parameters whose count does not increase as the number of data points increase.

Nonparametric methods do not summarize data into a fixed number of parameters. The number of parameters (if they're used at all) increases as the number of data points increase.

9. The Curse of Dimensionality (6 Points).

Describe the Curse of Dimensionality (esp. as described by Pedro Domingos in "A Few Useful Things to Know about Machine Learning").

The curse of dimensionality, coined by Bellman in 1961, refers to the fact that many algorithms that work fine in low dimensions become intractable when the input is high-dimensional. But in machine learning it refers to much more. Generalizing correctly becomes exponentially harder as the dimensionality (number of features) of the examples grows, because a fixed-size training set covers a dwindling fraction of the input space. Moreover, the similarity-based reasoning that machine learning algorithms depend on (explicitly or implicitly) breaks down in high dimensions.

10. Supervised vs. Unsupervised Learning (4 Points).

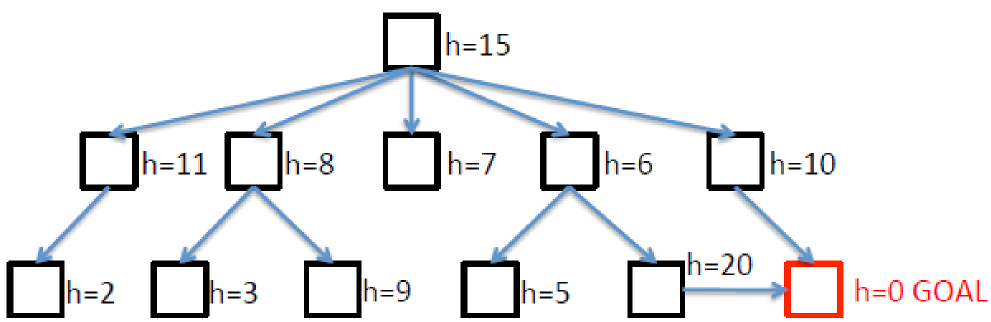
Describe the difference between Supervised and Unsupervised Learning.

Supervised Learning requires data that are labeled with their class during learning. Only class membership (classification) is determined by the supervised learning method.

Unsupervised Learning does not require data that are labeled with their class during learning. Both classes and class membership are determined by the unsupervised learning method.

11. A* Search (20 Points).

For heuristic function h and action cost 10 (per step), enter into each node the order (1,2,3,…) when the node is expanded (=removed from queue). Start with "1" at start state at the top. Enter "0" if a node will never be expanded.



1) The node where h=15: 1

2) The node where h=11: **0**

3) The node where h=8: **4**

4) The node where h=7: **3**

5) The node where h=6: **2**

6) The node where h=10: **5**

7) The node where h=2: **0**

8) The node where h=3: **0**

9) The node where h=9: **0**

10) The node where h=5: **0**

11) The node where h=20: **0**

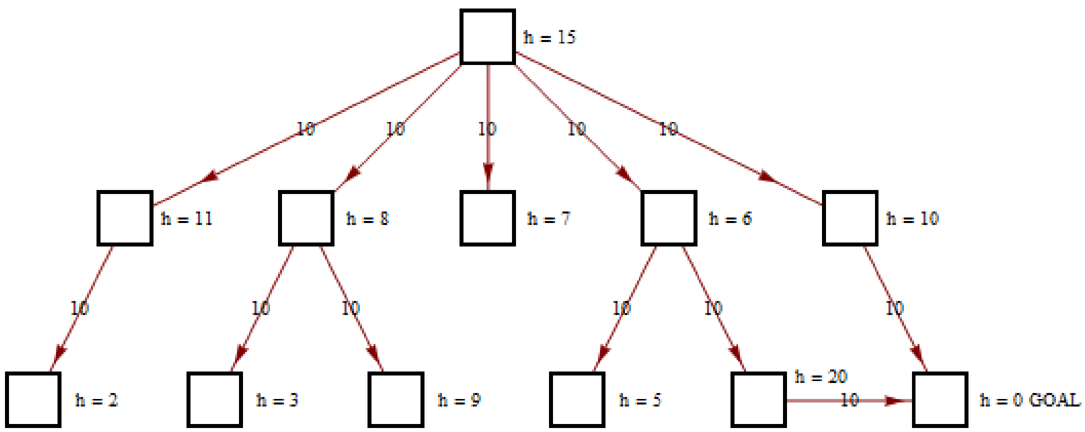
12) The node where h=0 GOAL: **6**

13) Is this heuristic admissible? If yes, why? If no, why?

No, the heuristic is not admissible because h=20 is an overestimate since single steps only cost 10 and the node where h=20 is only one step away from the goal.

Here's the exhaustive answer:

First, we augment the graph above with edge weights:



Next, we add the top *node* to the *frontier*:

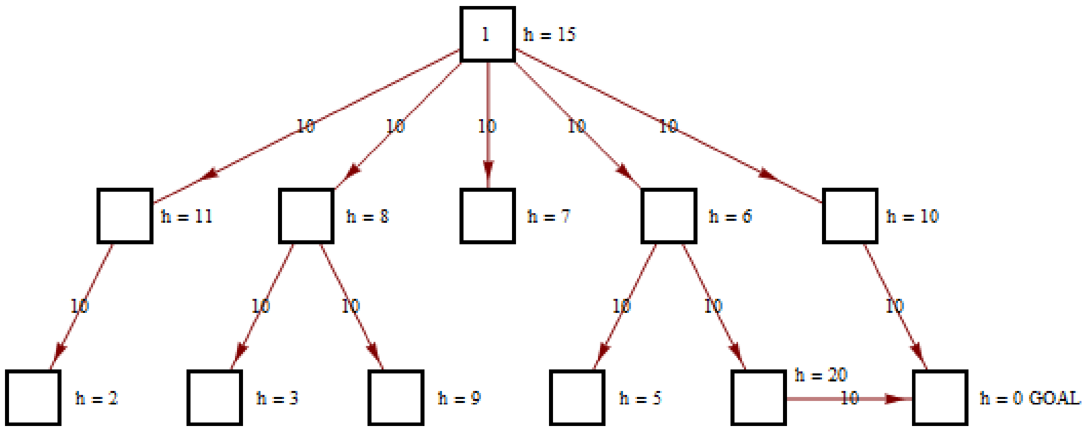
$$frontier = \left\{ \begin{pmatrix} f(n) = 15 \\ g(n) = 0 \\ h(n) = 15 \end{pmatrix} \right\}$$

and *explored* is empty:

$$explored = \{ \ \}.$$

Entering the loop, we see that *frontier* is not empty, so we pop *frontier* to *node*:

$$node = \begin{pmatrix} f(n)=15 \\ g(n)=0 \\ h(n)=15 \end{pmatrix}$$



Then, we test if *node* is the goal. Since it isn't, we add *node* to *explored*:

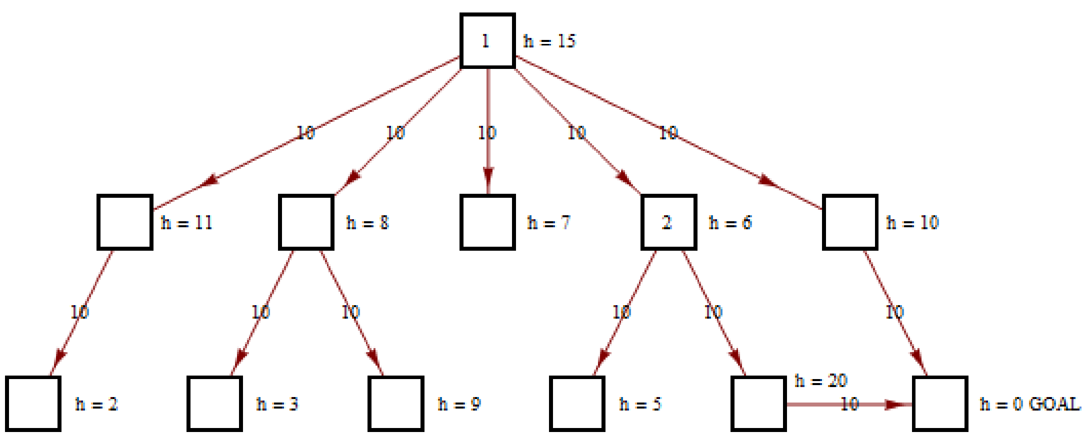
$$explored = \left\{ \begin{pmatrix} f(n)=15 \\ g(n)=0 \\ h(n)=15 \end{pmatrix} \right\}.$$

For each *child* of the *node*, we test if it is in *explored* and *frontier*. Since they aren't, we add each *child* to the *frontier* giving priority to the smallest $f(n)$:

$$frontier = \left\{ \begin{pmatrix} f(n)=16 \\ g(n)=10 \\ h(n)=6 \end{pmatrix}, \begin{pmatrix} f(n)=17 \\ g(n)=10 \\ h(n)=7 \end{pmatrix}, \begin{pmatrix} f(n)=18 \\ g(n)=10 \\ h(n)=8 \end{pmatrix}, \begin{pmatrix} f(n)=20 \\ g(n)=10 \\ h(n)=10 \end{pmatrix}, \begin{pmatrix} f(n)=21 \\ g(n)=10 \\ h(n)=11 \end{pmatrix} \right\}$$

On the next iteration, we again see that *frontier* is not empty, so we pop frontier to *node*:

$$node = \begin{pmatrix} f(n)=16 \\ g(n)=10 \\ h(n)=6 \end{pmatrix}$$



Since this *node* isn't the goal, we add the *node* to *explored*:

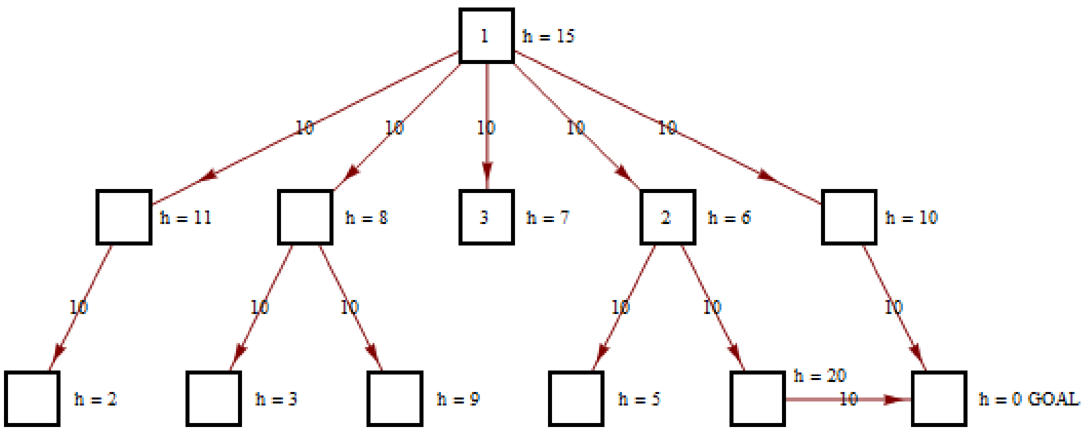
$$explored = \left\{ \begin{pmatrix} f(n)=15 \\ g(n)=0 \\ h(n)=15 \end{pmatrix}, \begin{pmatrix} f(n)=16 \\ g(n)=10 \\ h(n)=6 \end{pmatrix} \right\}$$

and then we look at the children of that *node*. For each *child*, we check to see if the *child* is in the *frontier* or in *explored*. Since each *child* isn't, we add these children to *frontier*:

$$frontier = \left\{ \begin{pmatrix} f(n)=17 \\ g(n)=10 \\ h(n)=7 \end{pmatrix}, \begin{pmatrix} f(n)=18 \\ g(n)=10 \\ h(n)=8 \end{pmatrix}, \begin{pmatrix} f(n)=20 \\ g(n)=10 \\ h(n)=10 \end{pmatrix}, \begin{pmatrix} f(n)=21 \\ g(n)=10 \\ h(n)=11 \end{pmatrix}, \begin{pmatrix} f(n)=25 \\ g(n)=20 \\ h(n)=5 \end{pmatrix}, \begin{pmatrix} f(n)=40 \\ g(n)=20 \\ h(n)=20 \end{pmatrix} \right\}$$

On the next iteration, we see that *frontier* is not empty, so we pop *frontier* to *node*:

$$node = \begin{pmatrix} f(n)=17 \\ g(n)=10 \\ h(n)=7 \end{pmatrix}$$



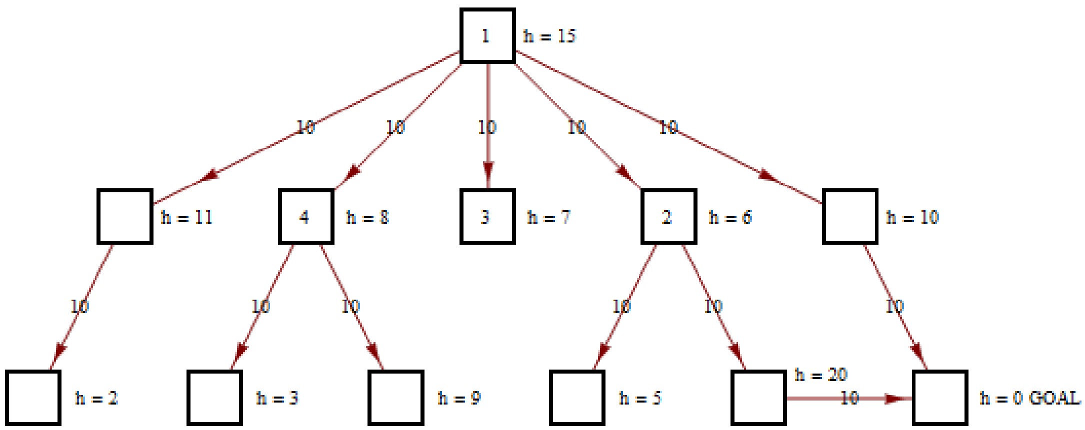
Since *node* isn't the goal, we add *node* to *explored*:

$$explored = \left\{ \begin{pmatrix} f(n)=15 \\ g(n)=0 \\ h(n)=15 \end{pmatrix}, \begin{pmatrix} f(n)=16 \\ g(n)=10 \\ h(n)=6 \end{pmatrix}, \begin{pmatrix} f(n)=17 \\ g(n)=10 \\ h(n)=7 \end{pmatrix} \right\}$$

Since *node* doesn't have any children, we go on to the next iteration.

On this next iteration, we see that *frontier* is still not empty, so we pop *frontier* to *node*:

$$node = \begin{pmatrix} f(n)=18 \\ g(n)=10 \\ h(n)=8 \end{pmatrix}$$



Since *node* isn't the goal, we add *node* to *explored*:

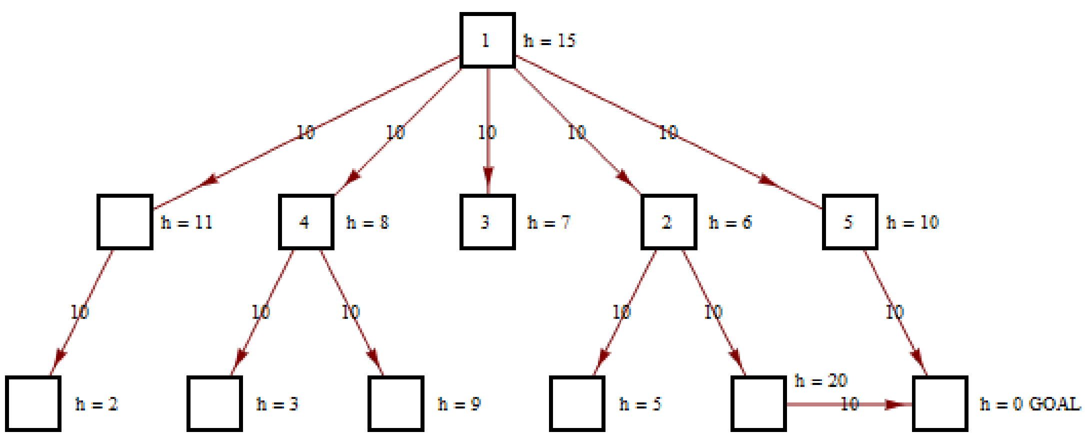
$$explored = \left\{ \begin{pmatrix} f(n)=15 \\ g(n)=0 \\ h(n)=15 \end{pmatrix}, \begin{pmatrix} f(n)=16 \\ g(n)=10 \\ h(n)=6 \end{pmatrix}, \begin{pmatrix} f(n)=17 \\ g(n)=10 \\ h(n)=7 \end{pmatrix}, \begin{pmatrix} f(n)=18 \\ g(n)=10 \\ h(n)=8 \end{pmatrix} \right\}$$

For each *child* of *node*, we check to see if the *child* is in the *frontier* or in *explored*. Since each *child* isn't, we add these children to *frontier*:

$$frontier = \left\{ \begin{pmatrix} f(n)=20 \\ g(n)=10 \\ h(n)=10 \end{pmatrix}, \begin{pmatrix} f(n)=21 \\ g(n)=10 \\ h(n)=11 \end{pmatrix}, \begin{pmatrix} f(n)=23 \\ g(n)=20 \\ h(n)=3 \end{pmatrix}, \begin{pmatrix} f(n)=25 \\ g(n)=20 \\ h(n)=5 \end{pmatrix}, \begin{pmatrix} f(n)=29 \\ g(n)=20 \\ h(n)=9 \end{pmatrix}, \begin{pmatrix} f(n)=40 \\ g(n)=20 \\ h(n)=20 \end{pmatrix} \right\}$$

On the next iteration, we see that *frontier* is not empty, so we pop *frontier* to *node*:

$$node = \begin{pmatrix} f(n)=20 \\ g(n)=10 \\ h(n)=10 \end{pmatrix}$$



Since *node* isn't the goal, we add *node* to *explored*:

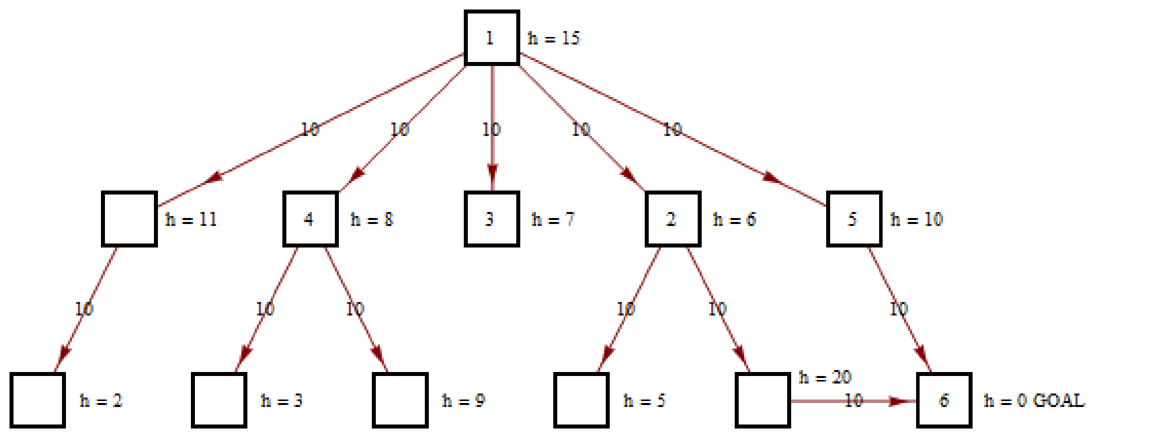
$$explored = \left\{ \begin{pmatrix} f(n)=15 \\ g(n)=0 \\ h(n)=15 \end{pmatrix}, \begin{pmatrix} f(n)=16 \\ g(n)=10 \\ h(n)=6 \end{pmatrix}, \begin{pmatrix} f(n)=17 \\ g(n)=10 \\ h(n)=7 \end{pmatrix}, \begin{pmatrix} f(n)=18 \\ g(n)=10 \\ h(n)=8 \end{pmatrix}, \begin{pmatrix} f(n)=20 \\ g(n)=10 \\ h(n)=10 \end{pmatrix} \right\}$$

and then we look at the children of that *node*. For each *child*, we check to see if the *child* is in the *frontier* or in *explored*. Since each *child* isn't, we add these children to *frontier*:

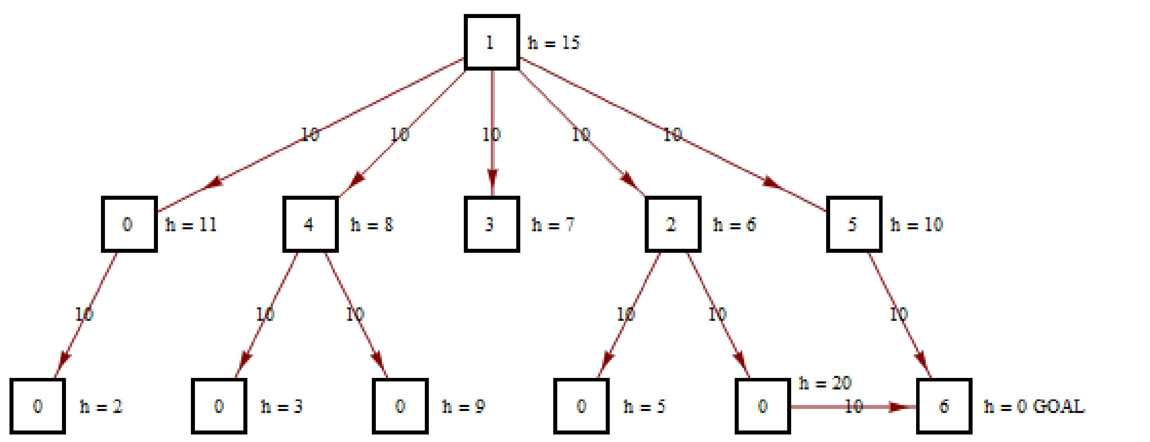
$$frontier = \left\{ \begin{bmatrix} \begin{pmatrix} f(n)=20 \\ g(n)=20 \\ h(n)=0 \end{pmatrix}, \begin{pmatrix} f(n)=21 \\ g(n)=10 \\ h(n)=11 \end{pmatrix}, \begin{pmatrix} f(n)=23 \\ g(n)=20 \\ h(n)=3 \end{pmatrix}, \begin{pmatrix} f(n)=25 \\ g(n)=20 \\ h(n)=5 \end{pmatrix}, \begin{pmatrix} f(n)=29 \\ g(n)=20 \\ h(n)=9 \end{pmatrix}, \begin{pmatrix} f(n)=40 \\ g(n)=20 \\ h(n)=20 \end{pmatrix} \end{bmatrix} \right\}$$

On the next iteration, we see that *frontier* is not empty, so we pop *frontier* to *node*:

$$node = \begin{pmatrix} f(n)=20 \\ g(n)=20 \\ h(n)=0 \end{pmatrix}$$



Since *node* is the goal, then we have our solution … and we're done … and we put zeros in the nodes that will never be expanded:

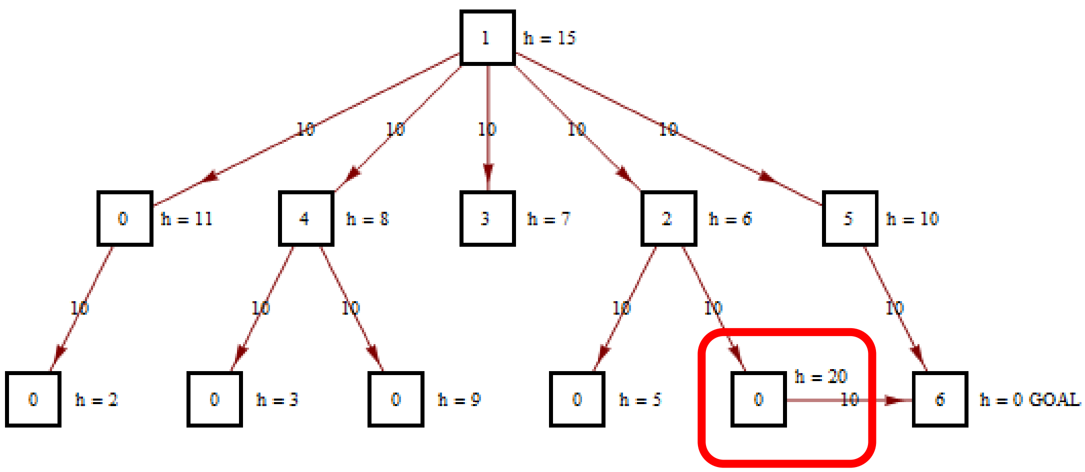


Therefore, we have the following:

a. The node where h=15: 1
b. The node where h=11: 0
c. The node where h=8: 4
d. The node where h=7: 3
e. The node where h=6: 2
f. The node where h=10: 5
g. The node where h=2: 0
h. The node where h=3: 0
i. The node where h=9: 0
j. The node where h=5: 0
k. The node where h=20: 0
j. The node where h=0 GOAL: 6

l. Is this heuristic admissible? If yes, why? If no, why?

Highlighting the following:



We see that our heuristic has a higher value than the actual edge cost to the goal because $\text{edge cost} = 10 < 20 = \text{heuristic}$; hence, our heuristic overestimates the cost to reach the goal. Therefore, since our heuristic overestimates the cost to reach the goal, then this heuristic is not admissible.

In other words: No, our heuristic is not admissible because it overestimates the cost to reach the goal.