# COMP 7745/8745: Machine Learning

## Instructor: Deepak Venugopal

## Spring 2018: Homework 2

## Due Date: March 12, 2018 (Hard copy or Ecourseware)

1. Naive Bayes (15 points). Given the below dataset,

| **W** | **X** | **Y** | **Class** |
|---|---|---|---|
| T | T | T | T |
| T | F | T | F |
| T | F | F | F |
| F | T | T | F |
| F | F | F | T |

   - Use Naive Bayes to classify the test example, (T, T, F).
   - Suppose I tell you that there are 100 more examples with Class = T, how will your classification change?
   - Can Naive Bayes handle noisy data? That is, in two different training examples, the features values are exactly the same but the labels are different? Briefly explain.

2. Does regularization change the expresiveness of logistic regression? Briefly explain. (10 points)

3. Given a training dataset D, is the 1-Nearest Neighbor algorithm guaranteed to have 100% accuracy on D. How about 3-Nearest Neighbor? Give a brief explanation of your answer. (10 points)

4. In your own words, explain the distinction between Bayesian learning, MAP learning and Max-Likelihood learning? (10 points)

5. Which of the following is easier to learn (in terms of sample complexity), 1-Nearest Neighbor or logistic regression? (10 points)

6. For the hypothesis class learned by perceptrons with $k$ inputs and a bias, what is the sample complexity to ensure with 95% confidence that the true error is at most 0.01. (10 points)

7. Consider a class of concepts of the form: $a \leq x \leq b \wedge c \leq y \leq d$. What is the sample complexity to assure with 90% confidence that the true error is at most 0.01, when assuming that each of $a$, $b$, $c$ and $d$ are represented using 8 bits. (10 points)

8. Consider a class of concepts of the form: $a \leq x \leq b$ . What is the sample complexity to assure with 90% confidence that the true error is at most 0.01, when $a$, $b$ are arbitrary real numbers with infinite precision. (10 points)

9. In this question, you will experiment with scalability of logistic regression, Naive Bayes classifier and K-NN (K=3) implementations in Weka (or other packages of your choice). Plot a graph that shows number of features on one axis (experiment with num-features = 5,10,15, 20 and 30), vs time taken for 5-fold cross validation on the other for a fixed data-size (number of training instances = 1000). Generate the data yourself (can just be random numbers) with the required number of features. Generate a second graph by fixing the number of features (num-features = 10) and varying data-size on one axis vs time taken for 5-fold cross validation on the other (experiment with number of training instances =100, 500, 1000, 10000, 25000). If you wish to automate this experiment, you can look at the sample-code wekarun.java on how to call the weka-libraries from Java (you need to have weka.jar in your classpath). (15 points)