Before going over the quiz questions, I’d like to discuss the goals of the course and how the quizzes fit into achieving those goals. The goals are to learn the basic principles of machine learning, to get experience with a variety of learning methods and algorithms, to understand the advantages and disadvantages of various methods in order to know which to apply in any given situation, to understand the design of learning algorithms, to prepare for more advanced machine learning courses, to be able to read current literature, and to be able to explain these concepts orally and in writing. A bit more abstractly, there are also the goals of learning how to study a new topic, how to take advantage of various resources, how to know when there is a gap in your knowledge, how to fill such a gap, and how to communicate questions and ideas to others.

These goals are not well served by the standard midterm/final exam structure, which encourages memorizing a lot of facts in short-term memory in order to get a good grade on exams. The point of having smaller random quizzes throughout the semester is to take the pressure off of large exams and to provide motivation to study well throughout the semester. So how can you study well?

1) Read the assigned material (book sections, slides, etc.) in advance of class and note questions that you might want to ask.
2) Pay attention in class, take good notes, and ask questions if something is unclear. Even if you’re not inclined to take notes, and even if you never look at them after class, it’s been demonstrated that the process of taking notes helps you absorb the material as you are hearing it.
3) If you don’t have an opportunity to ask a question in class, or if you have a question before or after class, take advantage of Piazza, email, TA hours, or request a meeting with the instructor.
4) Read the material again after class to see if you have a new perspective on it.
5) Discuss the material with others in the class.

A note on grades: I will often give 9/10 for a basic correct answer, reserving 10/10 for answers that are particularly good. 8/10 would mean a slight deficiency, with 7/10 and below reserved for more major misunderstandings.

1) (10 points) The \(k\)-means algorithm has one obvious parameter, \(k\), but other things could be modified to change how it clusters a given data set, without modifying the data set or the basic outline of the algorithm. Describe a few of these.

The following three were obvious to many people from working on the homework, but somewhat uninteresting:

a) Changing the way the initial centers were assigned
b) Changing the way a new center was assigned if a cluster became empty
c) Changing the stopping criterion from “stop if the error reduction is below 1%”

The following are more interesting, as they could lead to better clusters or new types of clustering:

d) Scaling different attributes differently. Note that this can be done without modifying the data, since different coefficients can multiply different attributes in the distance calculation
e) Rather than having \(0/1\) assignments of points to clusters, allowing points to belong fractionally to several clusters
f) Using a different distance function, such as Manhattan distance
g) Using a different error function, such as “sum of distances”

Note that these last two would require a new proof of convergence.
2) (10 points) In \( k \)-means, suppose you limit the number of clusters to at most \( \log_2(n^2) \), where \( n \) is the number of data points. How does the worst-case running time depend on \( n \), assuming that the dimension and the number of random restarts don’t change as \( n \) increases, and the number of iterations is also bounded by a constant independent of \( n \).

Each cluster center must be compared to every data point, so the running time is proportional to \( cn \), where \( c \) is the number of centers. \( c \) is at most \( \log(n^2) = 2 \log n \), so the worst-case running time is proportional to \( n \log n \).

3) (10 points) \( k \)-nearest neighbor is often called a lazy algorithm. Explain what this means, and discuss the advantages and disadvantages of this laziness.

This is due to the fact that \( k \)-nearest neighbor just stores all the data points until it needs to make a classification decision. This is very efficient as the data is coming in, but means that classification takes much longer, since the distances from every data point to the point to be classified must be calculated at that time.

4) (10 points) Suppose you are using \( k \)-nearest neighbor with \( k = 5 \) on a dataset with approximately 4% classification error. If you decided to switch to a similar dataset with an 8% classification error, would you consider increasing or decreasing the value of \( k \)? Explain.

If the error rate goes up, the probability of misclassifying a point goes up. It may be advantageous to increase the value of \( k \), which allows more points to vote on the classification, thus reducing the chance of error.

Quiz statistics:

Average = 24/40
Average on question 1: 9/10, question 2: 7/10, question 3: 4/10, question 4: 4/10