

## Quiz Submissions - Quiz 4 - Attempt 1



Nhat Le (username: hung.le@mail.mcgill.ca)

### Attempt 1

Written: Feb 13, 2019 11:19 PM - Feb 13, 2019 11:56 PM

### Submission View

Released: Jan 23, 2019 11:59 PM

View the quiz answers.

### Question 1

1 / 1 point

Suppose you are running a kernel-based (i.e., weighted) nearest neighbor predictor with the following weighting function:

$$w(d(\mathbf{w}_i, \mathbf{w})) = e^{-Cd(\mathbf{x}_i, \mathbf{x})}$$

Suppose we **increase** the value of  $C$  in this weighting function. Which of the following statements is generally true?

- ☐ The nearest-neighbor predictor converges towards estimating the mean value in the training data.
- ☐ The bias of the nearest-neighbor predictor increases and the variance decreases.
- ☒ The variance of the nearest-neighbor predictor increases and the bias decreases.
- ☐ The accuracy of the predictor improves.

▼ [Hide Feedback](#)

Increasing  $C$  is equivalent to decreasing

$\sigma$

from the standard Gaussian kernel in lectures (Lecture 9, slide 41-44). Thus, increasing  $C$  will decrease the width of the kernel and down-weight far-away neighbors, which is analogous to decreasing the number of nearest neighbors used---leading to a higher-variance predictor.

### Question 2

1 / 1 point

True or False: Naive Bayes is an example of a lazy learning algorithm.

- ☐ True
- ☒ False

▼ [Hide Feedback](#)

Naive Bayes works by learning a set of parameters during training time that summarize the data. It thus falls into the category of "eager learners" (see Slide 45 in Lecture 9).

### Question 3

1 / 1 point

You are computing tf-idf scores for a document,  $D$  within a larger training corpus  $C$ . Suppose you have three words---*the*, *dog*, and *follow*---with the following count statistics:

- *the* occurs 10 times in the document  $D$ , and, in total, *the* occurs in 50 documents in the corpus,  $C$ .
- *dog* occurs 5 times in the document  $D$ , and, in total, *dog* occurs in 30 documents in the corpus,  $C$ .
- *follow* occurs 2 times in the document  $D$ , and, in total, *follow* occurs in 2 documents in the corpus,  $C$ .

Further suppose that the corpus  $C$  contains 100 documents. Compute the tf-idf scores for these words. What would be their ranking (in descending order of tf-idf scores):

☒ follow, the, dog

☐ dog, follow, the

☐ the, follow, dog

☐ follow, dog, the

▼ [Hide Feedback](#)

Note that we are not given the number of words in  $D$ , so we cannot compute relative frequencies. However, this does not impact the ranking and we know that the tf-idf is proportional to the word count in the document multiplied by the idf score (Lecture 8, Slide 16).

Thus we have that

$$tfidf(\text{the}) \propto 10 \times \log \frac{100}{50 + 1} \approx 6.733$$

$$tfidf(\text{dog}) \propto 5 \times \log \frac{100}{30 + 1} \approx 5.856$$



$$tfidf(\text{follow}) \propto 2 \times \log \frac{100}{2 + 1} \approx 7.013$$

### Question 4

0 / 1 point

Suppose we have a dataset with  $m$ -dimensional continuous (i.e., real valued) features. Now, suppose that we train a **full** decision tree (Model A) using the C4.5 algorithm on this dataset, where we use binary threshold-based tests at every node, and suppose that we grow the decision tree until every leaf contains exactly one training point and perform no post-pruning. Suppose also that we train a one-nearest neighbor predictor (Model B) on this dataset.

Which of the following statements is most applicable to this situation:

-  ☐ Since this decision tree contains one training example per leaf, Model A will always make the same predictions as Model B.
-  ☐ Model A (the decision tree) may or may not make the same predictions as Model B (the one-nearest neighbor approach). It depends on the tests used in the decision tree and the distance function used in the nearest neighbor algorithm.
- ☐ Since it contains one leaf per training point, this decision tree is essentially a "lazy" algorithm.

 [Hide Feedback](#)

First off, the decision tree will not necessarily make the same predictions as the one nearest neighbor approach, because this is completely dependent on the distance function used in the nearest neighbor computation and the tests used for the decision tree.


Also, we cannot call this decision tree "lazy" because it is still generalizing before seeing a query. That is, even though every leaf corresponds to one training point, we are not actually storing these training points, we are only storing the information about the binary tests and the class of the points at each leaf, which is (in principle) less information than storing the entire training data. For instance, we cannot necessarily reconstruct the training data from the learned decision tree.

## Question 5

1 / 1 point

You plan to train a linear model **using gradient descent**. You are trying to decide between running PCA as a pre-processing step and then running regular gradient descent or not performing PCA and using L2-regularized gradient descent when training the model. Suppose your dataset has  $n=100$  training points,  $m=30$  dimensional features, and you plan to run  $I=1000$  iterations of gradient descent. When using PCA, you plan to reduce the dimensionality of the features to  $m'=10$ .

Which approach would you expect to have better time complexity (in big-O terms)? Assume that you are computing the PCA by using an eigendecomposition approach, with a time complexity of  $O(nm^2 + m^3)$  and that running (regularized) linear regression with gradient descent costs  $O(nmI)$ . You can ignore constant factors/terms in the complexity calculations and simply rely on the big-O terms above.

-  ☒ It would be faster to run PCA as a pre-processing step.
- ☐ It would be faster to simply run regularized linear regression.
- ☐ The methods would have the same time complexity.

 [Hide Feedback](#)

The big-O complexity of running gradient descent with L2 would be:

$$O(mnI) = 30 \times 100 \times 1000 = 3000000$$

The big-O complexity of running PCA followed by gradient descent would be:

$$O(nm^2 + m^3 + nm'I) = 100 \times 30^2 + 30^3 + 100 \times 10 \times 1000 = 1117000$$

---

**Attempt Score:** 4 / 5 - 80 %

**Overall Grade (highest attempt):** 4 / 5 - 80 %

Done