CS 374 Assignment #4

k-Nearest Neighbor and Social Implications of Machine Learning

Due the week of March 15, 2021

—

This week we explore another type of classifier learning algorithm: k-nearest neighbor. We also confront some negative consequences of machine learning.

1 k-Nearest Neighbor

The learning techniques we have seen so far (perceptron learning, Naive Bayes, logistic regression, top-down induction of decision trees) have involved the analysis of a training set of examples in order to build a model for the task at hand. New examples were then classified using the learned model. We now consider the k-nearest neighbor algorithm, which does not explicitly build a model from training data. Instead, it stores the training instances so that they can be used to analyze new instances later.

As its name implies, the k-nearest neighbor algorithm searches the training examples to find those that are "closest" to a new example to be classified. It then uses these to determine the appropriate output for the new instance. k-nearest neighbor is an example of a class of learning techniques called instance-based methods. These methods are sometimes referred to as "lazy", because they put off the processing of examples until a new test instance is considered.

1.1 Reading

Begin by reading

- Mitchell, pages 230-236.

This gives a very nice introduction to the k-nearest neighbor algorithm, with a focus on real-valued attributes. It discusses the advantages and disadvantages of this learning approach and outlines solutions for handling some of the problems. For example, it points out that not all neighbors of a new instance should be weighted equally in determining the output value for the new instance. It also points out that the process of finding the nearest neighbors of a new instance can be costly.

Next you should read

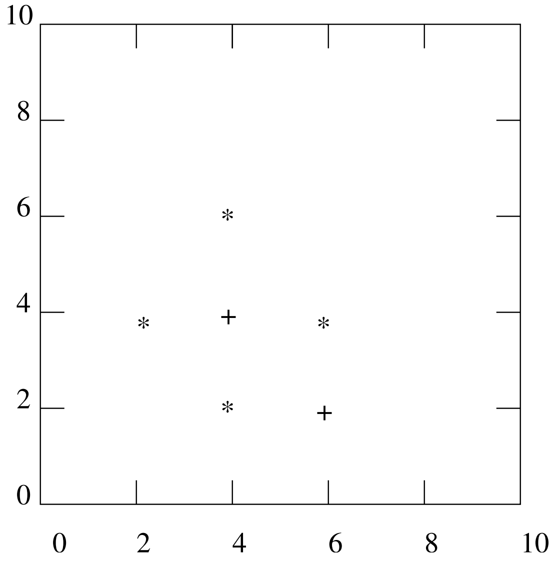
- Russell and Norvig, pages 737-741, and
- Witten and Frank, pages 78-79 and pages 128-135.

The reading in Russell and Norvig provides alternate ways to determine nearest neighbors, as does the reading in Witten and Frank. (Both, for instance, discuss how we handle nominal, i.e., discrete, attributes.) They also describe (at a very high level) data structures that can be used for storing training instances so that nearest neighbors of a new instance can be efficiently computed.

1.2 Exercises

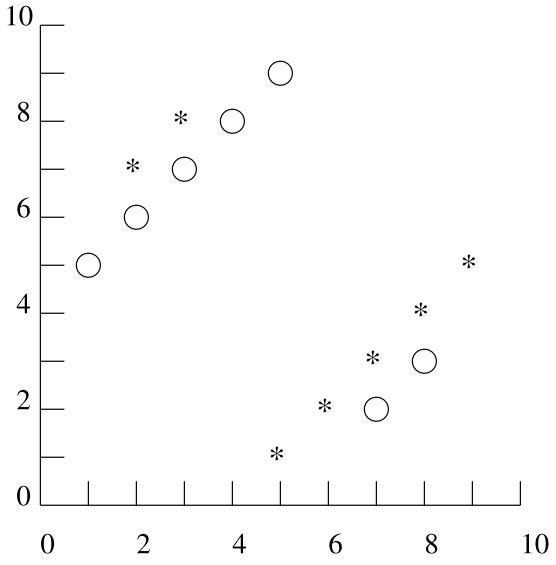
I will expect you to present k-nearest neighbor in tutorial. At a high level, how does it work? How are "nearest neighbors" defined for real-valued attributes? for nominal-valued attributes? Are there problems with treating all attributes the same? Are there ways to overcome this? How might one select k? Write up and be prepared to present all of the exercises below.

1. (Adapted from Dietterich) Consider the set of training examples in the diagram below.
   (a) Draw the decision boundaries for the 1-nearest neighbor algorithm assuming that we are using standard Euclidean distance to compute nearest neighbors. A plus indicates a positive example and a star indicates a negative example.
   (b) How will the point (8, 1) be classified by the 1-nearest neighbor classifier?
   (c) How will the point (8, 8) be classified?



2. (Modified from Mitchell and Guestrin) One of the problems with k-nearest neighbor learning is selecting a value for k. For this exercise, you will use Weka to empirically determine a reasonable value for k, given a specific training set.

   Say you are given the data set shown below. This is a binary classification task in which the instances are described by two real-valued attributes.



   (a) What value of k minimizes **training set** error for this data set, and what is the resulting training set error? (Part d of this exercise includes instructions for obtaining the data set and using the Weka implementation of k-nearest neighbors, though you **shouldn't need to use it here**.)

   (b) Why is training set error not a reasonable estimate of test set error, especially given the value of k chosen in part a)?

   (c) Why might using too large a value of k be bad for this data set? Why might using too small a value be bad for this data set?

   (d) Now find a value of k that minimizes leave-one-out cross-validation error. What we'll be doing here is using our training set to help us select a good value for k.

       The idea of using leave-one-out cross-validation should be a familiar one (recall the Naive Bayes assignment). In using it to select k, we do the following. We take our training instances and divide them into two groups, so that some are used for training and others serve as validation (i.e., "test") examples. Specifically, for the given training set of 14 examples, we divide it into a set of 13 for training and a set of 1 for testing. There are obviously 14 ways to choose 13 (i.e., "leave one out"), and we will consider all 14.

For each training-validation split of the data (remember that there are 14 of these), we run the k-nearest neighbor algorithm for all possible values of k (1 to 13). We then compute how well we did at classifying the validation instances for each value of k. The k that performs best is selected as the value to use for future (unseen) test instances.

To perform this experiment, you need to begin by copying the data file `week4.arff`, which can be found in

```
~andrea/shared/cs374
```

as well as in the Week 4 folder on Glow.

Then you'll run Weka. Recall from last week, that to run Weka's implementations of machine learning algorithms through the GUI, type the following in a terminal window when connected to one of the CS unix lab machines:

```
java -Xmx1g -jar /usr/share/java/weka.jar
```

or type the following on your own laptop:

```
java -Xmx1g -jar weka.jar
```

This week you will use the "Experimenter" function. Click on the "Experimenter" button. This should open a window for the Weka Experiment Environment. Here you can set up an experiment like the one described above.

First select "Simple" experiment, and then click "New" to set up a new experiment. Next you should type in the name of a file in which you would like to store the details of the experiment. Now under "Experiment Type", select "Cross-validation" and specify the number of folds to be 14. Also be sure that you've selected "Classification" under "Experiment Type."

The next thing you need to do is select the data set. In this case there's only one – `week4.arff`.

Under "Iteration Control", set the number of repetitions to 1, and select "Data sets first".

Finally, you need to specify the algorithms to be run, i.e., 1-nn, 2-nn, 3-nn, . . . , 13-nn. Click on "Add new", and then click on "Choose". In the "lazy" folder, select "IBk" (i.e., instance based k). Now you can edit the specifics of the algorithm. First make "KNN" 1. You should set "crossValidate" to False. You will need to do this 13 times.

Now you're ready to run the experiment. Click on "Run" at the top of the window. The experiment won't actually run until you click "Start", so do that now.

Finally, you can "Analyse" the results. To do this, click on "Analyse", and then click on "Experiment", which you'll find near the top right hand side of the window. The results of the experiment can be presented in many different ways. What you want is to see either the "Number correct" or the "Percent correct". You can choose this by selecting the "Comparison field". To actually see the results displayed in this way, click on "Perform test."

What value of k minimizes leave-one-out cross-validation error for this data set?

Play around with this interface for a bit, as you will be performing more experiments in the coming weeks.

3. (From Mitchell and Guestrin) A well-known result by Cover and Hart (1967) is that the asymptotic error rate of the 1-nearest neighbor classifier is at most twice the Bayes-optimal error rate. In this problem, you will prove Cover and Hart's theorem for the case of binary classification with real-valued attributes.

   Let $x_1, x_2, \ldots$ be training instances (points in d-dimensional Euclidean space, for some fixed d, and let $y_1, y_2, \ldots$ be the corresponding class labels ($y_i \in \{0, 1\}$). Let $p_i(x) = p(X = x \mid Y = i)$ be the conditional probability distribution for points in class i; we assume $p(x) > 0$ for all x. Let $R = p(Y = 1)$ be the probability that a randomly generated point is in class 1; we assume $0 < R < 1$.

   (a) From these expressions, we can easily calculate the true probability $q(x) = p(Y = 1 \mid X = x)$ that any data point x was generated by class 1. Express $q(x)$ in terms of $p_0(x)$, $p_1(x)$, and R.

   From this expression for $q(x)$, it is clear that $q(x)$ is defined for all x, and $0 < q(x) < 1$.

(b) A Bayes-optimal classifier is a classifier that always assigns a data point x to the most probable class, $argmax_y P(Y = y \mid X = x)$, thus minimizing the value of the 0-1 loss function, or equivalently, maximizing the probability of correct classification. Given some test point x, what is the expected error of the Bayes-optimal classifier, in terms of $q(x)$?

(c) The 1-nearest neighbor classifier assigns a test data point x the label of the closest training point $x\prime$. Given some test data point x with nearest neighbor $x\prime$, what is the expected error of the 1-nearest neighbor classifier, in terms of $q(x)$ and $q(x\prime)$?

(d) In the asymptotic case, the number of training examples of each class goes to infinity, and the training data fills the space in a dense fashion. As a result, the nearest neighbor to x has $q(x\prime)$ converging to $q(x)$. By performing this substitution in the previous expression, give the asymptotic error for the 1-nearest neighbor classifier at point x, in terms of $q(x)$.

(e) Show that the asymptotic error obtained in part d) is less than twice the Bayes-optimal error obtained in part b).

Please submit your solutions to the above problems as a single pdf.

2 Social Implications of Machine Learning

Interest in machine learning is widespread and growing. Though fundamental algorithms and concepts in machine learning have been explored for well over half a century, today's fast hardware, together with the ability to gather and store large quantities of data, make it possible to apply those algorithms toward the creation of new and exciting artifacts: cars that can drive themselves, personalized treatment plans for patients with cancer, brain-computer interfaces that allow paralyzed individuals to control the world around them, to name just a few. At the same time, our enthusiasm for learning from data can lead to unintended consequences, such as discrimination against certain groups of individuals. This may occur even in cases where there is no malice intended. It is simply too easy for some to assume that algorithms are objective and that the inferences they make from data must therefore be objectively correct.

This week you will explore work that has exposed bias in machine learned artifacts, where a key tool in the bias discovery process is itself machine learning.

2.1 Reading

Please read the following articles.

- "Apple Card algorithm sparks gender bias allegations against Goldman Sachs", Taylor Telford, The Washington Post, November 11, 2019.
- "When Computers Stand in the Schoolhouse Door", Neil Savage, Communications of the ACM, Vol 59, No. 3, March 2016.
- "Automated Experiments on Ad Privacy Settings", Amit Datta, Michael Carl Tschantz, and Anupam Datta, Proceedings on Privacy Enhancing Technologies, Vol. 1, 2015.
- "Certifying and removing disparate impact", Feldman et al., Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015.

2.2 Exercise

Write an essay – approximately 5 pages – responding to the following: A particular search engine mines users' searches for products, specifically focusing on those that they ultimately choose to purchase or reserve – e.g., hotels they choose to book for travel or household goods they choose to buy online. The mined information is then used by advertisers to differentially set prices that are high enough to be acceptable to the advertiser, yet low enough to entice a particular user to make a purchase. That is, prices advertised and set for an individual user reflect the buying power of that user. What are the implications (e.g., moral, legal) of such a system? Is it possible that this would violate antidiscrimination laws? Is there a process that could be applied to algorithmically determine possible bias?

Feel free to expand on these themes.

Please submit your essay as a pdf separate from the problem set above. For the tutorial session you should be prepared to present your essay and to discuss it with your partner and with me.