

## Project 4: Comparing Classification Results of Parametric Modeling versus Clustering on Two Data Sets: 1) Abalones and 2) Echocardiograms

---

**Assigned: Thursday, April 6**

**Due: Thursday, April 27**

*Note: this project is for individuals working alone. No teams on this project.*

---

**Introduction.** In this project, you will compare the performance of two types of learning approaches to classification on two different data sets. The two types of learning are (1) parametric learning and (2) clustering. The objective is to determine, through experimentation, the advantages and disadvantages of each of these types of learning for these data sets.

**Data Sets.** The data sets to use on this project are from the UCI Machine Learning Repository (<http://www.ics.uci.edu/~mlearn/MLSummary.html>). The two data sets we'll be using are the "Abalone" dataset and the "Echocardiogram" data set. These are described in more detail below.

The particular databases chosen for this project have numerical attributes, rather than discrete symbolic attributes, since we have been focusing on learning for numeric data using parametric learning and clustering. In most of the data sets on the UCI repository (including the 2 data sets to be used in this project), information is provided in two primary files: xxx.names and xxx.data (where xxx is replaced by the name of the database, such as "abalone" or "echocardiogram"). The xxx.names file contains documentation on the source of the data, past usages of the data (e.g., referencing specific research projects), some other descriptive information, and a definition of each of the attribute fields. The xxx.data file contains the actual data. You should closely read the xxx.names file for understanding which attributes are meaningful for learning (e.g., some of the attributes may give the classification of that data instance, so you'll use that to determine the class of your data instance rather than as a variable; or, an attribute could give a label for that data instance that isn't relevant for your learning process).

Abalone Database (<ftp://ftp.ics.uci.edu/pub/machine-learning-databases/abalone/>): Each instance of this data set contains physical measurements of an "abalone" (which is a type of mollusk – see picture). As it turns out, determining the actual age of an abalone requires cutting through the shell, staining it, and counting the number of rings under a microscope. Apparently this is not fun, and takes a long time. So, instead, people who care about abalones would rather predict the age of an abalone from physical measurements that are easier to gather (such as "length", "shucked weight", etc.). [Don't ask me what we can do with this age information after we've gathered it – I'm not a biologist! Maybe we can determine the health of the abalone population by ensuring that we have lots of abalones (abalene?) of all ages.] But, determining how to go from physical measurements to an age is not straightforward. So, this is where machine learning comes into the picture.

In this data set, ages range from 1 to 29. We could make each of these a separate class for our learning problem. But, it's a pretty good guess that our learning can't be this precise. So, instead, we'll group the ages into a smaller number of groups, and call each a class. In the abalone.names file, it mentions some prior work that created 3 classes from these ages, as follows: (class 1) ages 1-8, (class 2) ages 9-10, and (class 3) ages 11-29. So, we'll define 3 classes the same way.

Here are some details about this data set:

- 4177 instances
- 8 attributes. [Note that the first attribute represents the gender of the abalone, either "M" or "F". Since we are doing numeric estimation, you'll need to convert this to a numeric representation (e.g., 0 and 1). Note also that the last attribute represents the class (i.e., the correct answer for that data instance).]
- 29 classes (representing ages 1-29). [Note, again for your learning, you should reduce the number of classes to 3 as previously described.]
- No missing attributes

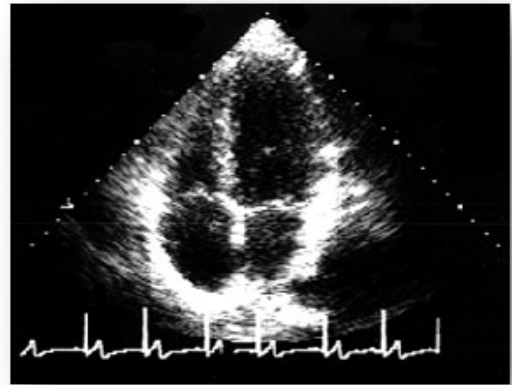


Echocardiogram Database (<ftp://ftp.ics.uci.edu/pub/machine-learning-databases/echocardiogram/>): This data set contains echocardiogram data (and other patient data) on a group of people who have had heart attacks in the past. In case you aren't familiar, an echocardiogram is a sonar test of your beating heart, which can be used to visualize and analyze the functioning of your heart. The picture below shows a snapshot taken during an echocardiogram; each of the 4 chambers of the heart is clearly visible. The objective of this data set is to try to use features of an echocardiogram (such as "left ventricular end-diastolic dimension" and "E-point septal separation" – again, don't ask me, I'm not a medical doctor, either ☺) to predict whether a person will still be living a year after their heart attack.

Some of the people represented in this data set are still alive (at the time the data set was created), and sadly, some are not. The machine learning objective is to predict whether or not a person will survive at least one year after a heart attack, based on features of the echocardiogram. That is, you want to classify the data into 2 classes: (class 1) patient dead after a year and (class 2) patient alive after a year.

Here are some details about this data set:

- 132 instances
- 13 attributes (but you won't use all 13)
- 2 classes (i.e., patient alive after 1 year, or patient dead after 1 year). [Pay attention to the notes on this database to be sure you are interpreting the data correctly. There are 3 attributes relating to this alive/dead issue. For instance, patients who are still alive at the time the data was collected, but who haven't been alive for at least 1 year since their heart attack, are not useful for this learning task. The data set notes clearly explain how to interpret these attributes.]
- Some missing attribute data. [Note that the missing attributes require you to pre-process the data to figure out what to do. You can do whatever you think is reasonable, but be sure to mention what you did and your reasons in your paper.]



### Learning Techniques to Apply, Data to Gather, and Comparisons to Make

In this project, you will apply 2 types of learning techniques – parametric learning (also known as density estimation) and clustering. Both techniques should be applied to both data sets. You will then compare and contrast the results. The following subsections outline this in more detail.

#### *Parametric learning*

Since these datasets are both multivariate, you will (surprise!) be using multivariate methods for estimating the parameters of a distribution, as outlined in Chapter 5 of your text. You will apply all of the following twice – once for each data set. For this learning technique, you need to do the following:

- Analyze whether the data is in fact multivariate normal, by plotting the data as two by two bivariate scatter plots (see page 102 of text). Alternatively, if you are familiar (or want to look up) other analytical tests for whether your data is in fact multivariate normal, you may apply those techniques. In your paper, you should state your conclusions on whether the data is in fact multivariate normal, along with data (or graphs) that support your conclusions. Regardless of whether or not you find the data to be multivariate normal, I still want you to apply the rest of the parametric learning techniques.
- We're going to assume your data is multivariate normal (even though it may not be). Now, estimate the parameters of your model (i.e., the mean vector  $\mu$  and the covariance matrix  $\Sigma$ ), per the details given in section 5.2 of your text.
- Build 4 alternative types of discriminant functions for your model, as outlined in Table 5.1 on page 99 and in the related text, and determine which function performs the best (may be different for each data set). Note that for each type of discriminant, you will have  $K$  functions  $g_i(\mathbf{x})$ , for  $i$  going from 1 to  $K$ , where  $K$  is the number of classes in your data set. Note also that these 4 alternative types of discriminant functions correspond to equations 5.25, 5.24, 5.23, and 5.20 (in order of increasing complexity). Some of these equations require you to compute the determinant and inverse of a matrix. Code (written in C, obtained from *Numerical Recipes in C*) to

make these calculations is provided for you in the directory `~parker/courses/cs594ml/Project4`, in files `matrixmath.c` and `matrix.h`. File `matrixmath.c` has example code showing you how to call the functions in `matrix.h` to calculate the determinant and inverse of a matrix.

You determine which of the 4 forms of discriminant function is best by using some (say, half) of your data for training, and the remainder for cross-validation. Plot the training error (which here is measured as the percentage of misclassified data instances on the training data) and the validation error (i.e., percentage of misclassified data instances on validation data) as a function of the type of discriminant function you are using. (This will be similar to Figure 4.7, except here the x-axis values will correspond to the 4 alternatives on page 99.) Remember, to classify a data instance, you evaluate each of the  $K$  discriminant functions and assign the data instance to the class  $i$ , where  $g_i(\mathbf{x})$  gives the maximum value on that data instance over all  $i$ .

Analyze your results and determine which of the 4 types of discriminant functions is the best for your data (again, this may be different for each data set), based on the errors you have measured.

- Compare and contrast your overall findings for applying parametric learning to your two datasets. If you found similar results on both data sets, explain why you think they should be similar. Or, if you found differences, explain why you think they were different.

### Clustering

- To evaluate your ultimate results, you need to first divide your data in half, with half being the training data and the other half being the validation data. Apply the k-means algorithm (Figure 7.3) to your training data, where  $k$  is the number of classes of your data. Here, you will ignore information on the actual cluster you are told that a data instance belongs to. However, for this first step, assume you do know the number of classes  $k$  (which is the same number of classes you used in the parametric learning). Once you have clustered the data, you need to give each cluster a class label. Here, you will do this by assigning each cluster the class value that occurs most frequently in its data members. Next, you need to calculate the classification error, which counts the number of times a data instance's actual class (from the data set) does not match its cluster label. Calculate the classification error of the result on (1) training data, and (2) validation data.
- NOTE: This part is for the abalone data set ONLY. For this data set, we previously selected the definition and number of classes somewhat arbitrarily (i.e., based on age ranges 1-8, 9-10, and 11-29). But, since this was arbitrary, maybe that wasn't a good choice. We'll use clustering to see if there is a better choice. So, now, assume you do not know the number of classes,  $k$ . The objective is to determine if there are natural clusters that correspond to certain age ranges (which might be different from what you used earlier). Re-run your k-means algorithm, starting with  $k=2$ , and increasing the value of  $k$  by 1 in each iteration. As you do this, plot the "reconstruction error" as a function of  $k$ , as defined by equation 7.3 on page 136 in your text. Determine the "elbow" of this curve, which shows when an increase in  $k$  yields only a small improvement in the reconstruction error. This gives you the preferred  $k$  (and also tells you when you can stop iterating on  $k$  ☺).

Now, determine how well these clusters correspond to age ranges, by using your known information about the age (i.e., based on the data set attribute). Assign age ranges to each cluster (based on the actual ages from your data set, selecting the most frequent ages as the cluster's age range) and determine if your clustering is meaningful for predicting ages. (E.g., you might find clusters, but these clusters aren't predictive of the age of the abalone, since the age ranges might overlap significantly from class to class or even be randomly distributed.) If age isn't a clear meaning behind the clusters, make a stab at speculating what other possible physical meaning might lie behind the clusters found (i.e., in the domain of abalones, about which I know all UT students are experts ☺, being the landlocked state that we are).

### Overall Comparison

- Compare and contrast the parametric learning results with the clustering approach, in terms of any issue you think is relevant (at least in terms of accuracy). Summarize all of your findings for each variant of learning for each data set in a single table (i.e., 1 table total for everything). This table will show the performance (and any other issue of comparison) for each method for each data set. Discuss which approach gives the best results over all methods studied, and why you think that approach is best for the given data set(s). Discuss differences (or lack thereof) in your findings for each dataset. Discuss whether you think the size of the data set affected the results. If you were given a different data set, and the choice of one of the learning approaches you studied, discuss which learning approach you would try first for that data, and why. Discuss any other interesting observations that you may have noticed during this learning exercise.

## Paper Guidelines

As always, as part of your completed project, you must prepare a paper (3-6 pages) describing your project. Your paper should be formatted using common word processing software (such as LaTeX or Word), and should include the following:

- An abstract of 200 to 300 words summarizing your findings.
- An introduction describing the learning task and your approach.
- An explanation of the results, as outlined in the previous section. Use additional figures, graphs, and tables where appropriate.
- A discussion of the significance of the results.

## Undergraduate Grading

Your grade will be based primarily on the quality of your project implementation and your description of your findings in the paper writeup. You should have a “working” software implementation, meaning that the learning algorithm is implemented in software, it runs without crashing, performs parametric density estimation and clustering, and is well-documented.

Since this problem requires less design skill, and focuses more on the implementation of well-defined statistical techniques, it is expected that your learning program(s) will work properly.

Additionally, you must proofread your paper, ensuring no spelling or grammatical errors (such errors will reduce your grade). Figures and graphs should be clear and readable, with axes labeled and captions that describe what each figure/graph illustrates.

## Graduate Grading

Graduate students will be graded more strictly on quality of the research and paper presentation. I expect a more thorough analysis of your results. Your analysis should include a discussion (and perhaps results) on the following points (in addition to the points previously noted above):

- An analysis of dimensionality reduction for this problem, perhaps even performing dimensionality reduction through subset selection or principal component analysis. If you do implement one of these techniques, discuss the results and their impact on the classification error.
- Analysis of computational and storage requirements of the alternative approaches, implementation issues (such as ease of implementation), etc.
- Future work that you believe would improve the learning
- Any other insightful observations you’d like to make

*You should not address these points in a bullet-type fashion, but instead work the answers into your paper in a discussion-style format. The paper should have the “look and feel” of a technical conference paper, with logical flow, good grammar, sound arguments, illustrative figures, etc. Graduate students are expected to format their paper in standard IEEE conference format (see <http://www.ieee.org/portal/pages/pubs/transactions/stylesheets.html> for style files).*

*However, even with this additional information, your paper must not exceed 6 pages.*

---

## WHAT TO TURN IN:

You should email BOTH your project and paper to BOTH the instructor AND the TA ([parker@cs.utk.edu](mailto:parker@cs.utk.edu); [m Bailey@cs.utk.edu](mailto:m Bailey@cs.utk.edu)) by the deadline.

Your submission should be in 2 parts (i.e., submitted in exactly 2 files):

1. Paper (in pdf format). Name this file *Yourlastname-Paper-4.pdf*
2. Tar or zip file (OK if it’s compressed) of the programs and data files needed to run your learning algorithm, including a README file that gives instructions on how to run your code, and (if relevant), a makefile for creating the executable of your code. Name this file *Yourlastname-Project-4.tar* (or *.zip*, or *.tar.gz*, etc.).