AN INTRODUCTION TO MACHINE LEARNING FOR STUDENTS IN SECONDARY EDUCATION

Steven D. Essinger, Gail L. Rosen

Drexel University
Department of Electrical & Computer Engineering
3141 Chestnut Street
Philadelphia, PA 19104

ABSTRACT

We have developed a platform for exposing high school students to machine learning techniques for signal processing problems, making use of relatively simple mathematics and engineering concepts. Along with this platform we have created two example scenarios which give motivation to the students for learning the theory underlying their solutions. The first scenario features a recycling sorting problem in which the students must setup a system so that the computer may learn the different types of objects to recycle so that it may automatically place them in the proper receptacle. The second scenario was motivated by a high school biology curriculum. The students are to develop a system that learns the different types of bacteria present in a pond sample. The system will then group the bacteria together based on similarity. One of the key strengths of this platform is that virtually any type of scenario may be built upon the concepts conveyed in this paper. This then permits student participation from a wide variety of educational motivation.

***Index Terms*—** Machine Learning, Pattern Recognition, Secondary Education, Lab Modules

1. INTRODUCTION

Machine learning (ML), a subfield of artificial intelligence, has evolved out of the need to teach computers how to automatically learn a solution to a problem. In engineering this field is referred to as pattern recognition, aptly named because the computer is extracting patterns out of data and making a decision based on the pattern identified. It is a rich field that is broadly and inherently related to signal processing most notably through data-driven learning methodologies [1, 2].

Our understanding of human learning has inspired many of the ML methods currently available. For example, take a look at the foundation of neural networks, which is based off the structure of the interconnection of multiple neurons from the brain [3]. While this class of methods undoubtedly employs coarse approximations of actual neuron function, they have shown tremendous success in several ML applications [4].

A few examples of ML applications include speech recognition aka natural language processing, image processing such as face detection, DNA sequence classification, financial analysis such as detecting credit card fraud, sports prediction and search engine algorithms which have been put into use by major household name search providers [5, 6, 7]. Many ML techniques do require a moderate mathematical background. Statistics, linear algebra, and calculus are commonplace in many of the algorithms. Fortunately, simple mathematical techniques have been shown to be quite successful on practical problems exploiting ML. With only an understanding of means and Euclidean distances, students can be shown how to instruct a computer to identify the difference between a pen and pencil, albeit with the proper assumptions. The simplicity of the math therefore permits accessibility of the field of ML to the high school student.

ML not only employs mathematics for practical purposes, but also demands problem-solving skill at a fundamental level since each problem encountered requires proper tool selection and then the interfacing of the tool to the problem. Through the application of lab modules such as the one proposed in this paper, students may be exposed to multi-interdisciplinary fields simultaneously such as engineering, mathematics, computer science and the field of the problem being addressed such as biology, economics, photography, etc.

2. BACKGROUND

There are innumerable examples with which machine learning techniques may be employed to facilitate automatic problem solving. Suppose you wish to separate quarters, nickels and dimes. What information would the computer need to distinguish between these three types of coins? Think about how you would do the task yourself. It is easy to see that each type of coin has a different size. So all we need to do is supervise

This research is based upon work supported by the National Science Foundation under Grant Nos. #0733284 and #0845827.

978-1-61284-227-1/11/$26.00 ©2011 IEEE

or rather tell the computer the circumference of each type of coin and then let the computer automatically do the sorting. This problem is exceedingly easy and does not necessarily require any complex ML technique. Now let's say that we have a bag of unknown coins of several different currencies. We wish the computer to sort the coins by type and currency automatically. In this case the computer must learn the types for sorting and then classify each coin without our explicit input as in the previous example. This problem requires an unsupervised solution in that we do not know the types of coins beforehand. This example will serve as the basis for delineating the basic steps involved in developing a machine learning solution as depicted in figure 1.

The first step in any engineering design is to define the problem. In our example our problem is to sort a bag of unknown coins. To further motivate the use of ML techniques lets say that the bag contains 1,000,000 coins so that a manual solution of sorting is unrealistic for all practical purposes. We cannot simply just feed the computer the bag of coins, but must provide it with some information that it can use to make a decision about each coins type or rather its class as its known in the ML literature. This second step of the design is known as feature extraction.

Feature extraction requires the user to provide the computer with information that may be used to differentiate the classes, which in our case is the types of coins. We need to insure that our features contain enough discriminatory information about the classes.

For example, shape would probably be a poor choice since each coin is assumed to be round. However, if some coins were non-circular than this could be one useful feature. Many times we include multiple features to aid the algorithm. For our example the diameter of the coin is probably a useful feature, but since we have coins of multiple currencies there is a good chance that two different types of coins will have the same diameter. Therefore we chose a second feature such color of the coin. But since we are working with mathematics and computation, we need to map color to a number. We cannot simply say silver or copper, but instead find a numerical quantity that furnishes the same discriminatory information, say RGB value or perhaps luster. Let us choose luster since its arguable easier and cheaper to obtain. Some of the coins may be older than the others and therefore may not be as lustrous as those newer of the same class. This variation could be viewed as each type of coin having a mean value of luster and a standard deviation of luster across all coins of that type. Depending on the chosen algorithm, this information could be of use for separating the coins. A third feature may be added such as weight of the coin and a fourth, albeit extreme example, such as the bacterial composition found on each coin with the assumption that the composition varies depending on the region of the world the coin originated from. From this discussion, it is therefore no surprise that if poor features are chosen, then the algorithm will perform poorly. This is known as the garbage-in, garbage-out theorem and is why feature selection is one of the most important steps in ML and is almost always carried out by the designer.

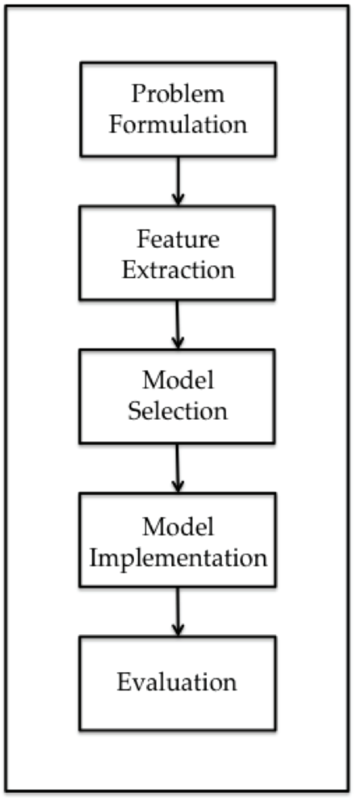

Fig. 1. This block diagram outlines the basic steps required to be addressed when developing a machine learning solution.

There are potentially hundreds, if not thousands of different ML algorithms to choose from. In many instances, the algorithms are modified to fit a particular problem resulting in a new model, thereby growing the population of choices. High school students should not have a problem comprehending Euclidean distance and mean so we suggest the use of the K-means algorithm for solving our coin sorting example and the subsequent lab activities. The details of the K-means algorithm are discussed in the next section. Of course, more advanced algorithms may be selected based on the students mathematical aptitude resulting in a highly scalable ML lab platform.

No matter which algorithm is chosen it must be properly coded and executed on the computer. Many programming options are available, but we suggest the use of Matlab for rapid implementation (www.mathworks.com). This software package already includes the K-means algorithm as a one-line command thereby alleviating the students or teacher from coding the algorithm from scratch. Alternatively, coding this simple algorithm could be an excellent exercise for a student in a computer science course using any language such as C, Java, Perl, Python, etc. Weka is another great machine learning tool for algorithm implementation that is based on Java [8, 9, 10]. The bottom line is that the coding part of the ML activity can be kept as simple or scaled as complex as desired by the educator without loss of generality.

Once the corresponding data has been captured based on the features selected and the algorithm has run on the data we evaluate the performance of the classifier on the dataset. In our example we expect to have groups of coins with each group containing one type of coin. In order to evaluate the performance of our ML selection we need to know the ground truth of the type of each coin. Here we may simply visually inspect the coins in each group. Under other circumstances this information could be recorded, set aside and kept unknown to the algorithm until after it has run, upon which the truth is compared to the algorithms output.

There are nearly as many evaluation metrics in ML as there are algorithms. The most appropriate metric depends on the specific application and problem being solved making the selection another aspect of design. We could simply count the number of misplaced coins and use that figure as our metric. A metric known as the rand index would be suitable for our ML design [11]. This metric shown in eqn. (1) essentially assesses the similarity of the groupings between the algorithms output and the ground truth. It is relatively straightforward to implement the rand index in Matlab or an Excel spreadsheet.

$$R = \frac{a+b}{a+b+c+d} \tag{1}$$

- a, the number of pairs of elements that are in the same set in X and in the same set in Y
- b, the number of pairs of elements that are in different sets in X and in different sets in Y
- c, the number of pairs of elements that are in the same set in X and in different sets in Y
- d, the number of pairs of elements that are in different sets in X and in the same set in Y
- X is referred to as the algorithm output and Y is the ground truth

3. K-MEANS ALGORITHM

The K-means algorithm is a very simple yet effective ML learning technique [12]. Besides the actual data (extracted features) from our coin example all we need to provide is the number of groups, K, we want the algorithm to produce. Ideally, we would want the algorithm to determine this number on its own, but in this case we would have to provide it with some other piece of information. This is known as the no free lunch theorem in that we must always specify at least one free parameter for all ML techniques. There are additional methods available to assist us in estimating the number of groups for this problem, but we will just assume that we now know the number of different types of currency in the coin bag, namely 3. We will chose the circumference and the luster of the coins as our features and run the K-means algorithm with K=3.

The steps for the K-means algorithm are as follows with a graphical depiction shown in figure 2:

1. Choose 3 staring points randomly. These are called the centroids.
2. Calculate the Euclidean distance between each point and centroids.
3. Assign each point to its nearest centroid.
4. Calculate the mean of each centroid based on the points assigned to it.
5. Move the centroid to the mean location.
6. Repeat steps 1-5 until centroids no longer move.

Each coin is represented as a 1 × 2 vector, $[x_1, x_2]$, with x_1 and x_2 corresponding to circumference and luster respectively. Given a set of objects $(x_1, x_2...., x_n)$, with each object represented by a d-dimensional vector, the k-means algorithm partitions the data into k sets S = $\{\mathrm{S}_1, S_2, S_3, ..., S_k\}$. The K-means algorithm aims to minimize the within-cluster sum of squares described by eqn. (2).

$$arg\ \underset{S}{min} \sum_{i=1}^{k} \sum_{x_j \epsilon S_i} \|\boldsymbol{x_j} - \mathbf{u_i}\|^2 \tag{2}$$

The students may perform the K-means algorithm by hand on a few instances by following steps 1-6 above using their knowledge of Euclidean distance (eqn. (3)) and mean (eqn. (4)).

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3}$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{4}$$

4. CLASS EXERCISES

The preceding information on ML and the K-means algorithm has been provided as a framework for customized activities for implementation in secondary education. There are essentially an unlimited number of exemplary problems that may

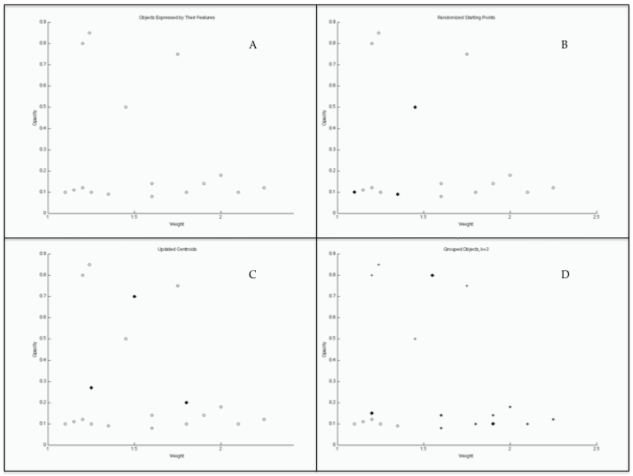

Fig. 2. This figure identifies several key steps in the k-means algorithm. Part A shows the objects represented by their features. Part B show the randomization assignment of centroids. Part C, displaying an intermediate step, depicts the movement of centroids based on the euclidean distance of each point and the centroids. Part D show the final location of the centroids and the assignment of points to their respective centroid.

be proposed to students varying with level of depth and complexity based on the student/teacher needs. The following examples are proposed as potential lab activities for implementation in the classroom.

4.1. Recycling Containers

The students have just been hired to design a container sorting system at their local recycling center. The system must be able to identify glass, plastic and cardboard drink containers so that they may be automatically placed in their respective bins. The students are to design and implement the artificial intelligence portion of the system using the ML techniques as delineated above.

A potential solution for this activity includes the layout of the system as shown in figure 3. Here the containers move along a conveyor belt where one by one their opacity is measured and then their weight is taken using the scale. These two features are stored in the variables x and y respectively. Once all of the containers have been analyzed the data is run through the k-means algorithm with k set to 3. The algorithm outputs three groups corresponding to the three different types of containers. Now when each container reaches the end of the conveyor belt the system automatically rotates the appropriate bin under the container based on its grouping as identified by the algorithm. The accuracy of this approach could be evaluated using one of the metrics described such as the rand index.

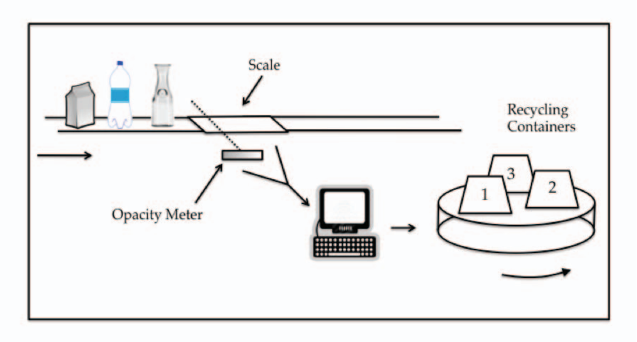

Fig. 3. This figure depicts a suggest solution for the recycling problem. Glass, plastic and cardboard items are placed down a conveyor belt towards the recycling bins. To extract their features they each pass through an opacity meter and a scale. The computer then runs the k-means algorithm to determine the appropriate bin for each item and rotates the platform accordingly.

4.2. Bacterial Classification

Several water samples have been obtained from a local pond as part of a biology lab series. The students in class have already been learning about plant and animal cells. Through their lab activities they have developed an appreciation of the labor involved with the identification and separation of the plant from animal cells in the pond samples. Since this is going to be an ongoing project they decide to use their newfound ML expertise to design a system to automatically identify the plant and animal cells present in each sample. To increase the difficulty of the problem we include a third type of bacteria in the sample such as Euglena, which has both animal and plant features (i.e. a chloroplast and flagellum).

A potential solution includes setting up a video monitor that projects the image from a microscope viewing a particular pond sample as shown in figure 4. The bacteria are each numerically labeled for tracking. The students decide to extract three features from each bacterium: shape, size and mobility. The data is entered in spreadsheet form and is submitted to the k-means algorithm. Since they know that there are two groups of bacteria, plant and animal, they decided to set k=2. The performance of the algorithm on this test dataset may be evaluated using any of the metrics discussed such as the rand index. It should be of interest to the students to see how the Euglena cells are grouped.

5. TOPICS FOR CLASS DISCUSSION

There are several areas for post-lab discussion that are beneficial regardless of the specific example and algorithm chosen. For instance, perhaps the students ran each experiment multiple times using different features. How did the features affect performance? Are some superior to others? Could all of the

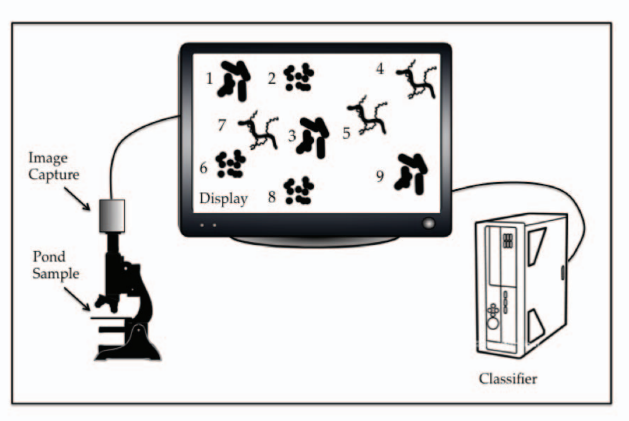

Fig. 4. This figure depicts a potential scenario for bacterial classification. The pond sample is placed under a microscope with a video sensor attached to the lens. This sensor is connected to a large monitor that displays the magnified pond sample. The data extraction from each object could be completed by the students or hypothetically by image processing software. Either way this data is input to the k-means algorithm. The output of the algorithm will specify to which group each bacterium belongs.

features be used? What about using more than three features? Can we visualize this data? Why or why not?

Each approach undoubtedly has its own strengths and weaknesses to be highlighted. In the recycling example the containers are all analyzed before each is deposited in its appropriate bin. How does this fair on the design of the conveyor belt? In the bacterial classification example the students had to deal with a bacterium that has both animal and plant features. How did this bacterium group? Was it consistent? Could they infer whether it is more of one class over the other? What about automating the feature extraction? Could they develop a ML technique to automatically measure the shape, size and motility of each bacterium?

Another important concept is that of generality. The students can run their ML design with different datasets, but remain using the same parameters and features. Based on the evaluation criteria does each run perform similarity or are some much better than others? What impact would this have if they were to spend a large sum of money on an unpredictable solution?

6. STRUCTURE OF LAB IMPLEMENTATION

We propose a variety of methods for presenting the lab to the students, although any combination of pedagogy may be chosen for a particular classroom. Our past experience designing and implementing signal processing labs on topics such as image processing and bioinformatics have taught us that the students tend to respond best to these topics when they are first briefed on the lab and background followed by a short, open class discussion [13]. In this ML lab we suggest class participation when walking through the example exercise such as the coin-sorting problem. The students may then break off into small groups to work on another exercise such as the bacterial classification problem. The results of each group's algorithm performance may then be compared and discussed collectively as a class. The lab could then conclude with one or more of the ideas from the suggested class discussion topics.

7. FUTURE WORK

The scalable nature of the proposed ML lab suggests that additional lab modules may be developed that expand on the basic concepts described here. We envision labs utilizing neural networks to highlight advanced ML techniques as well as provide insight into biologically inspired algorithms. We also intend to develop an activity for use in classrooms where students have chosen to focus on fields stemming from the creative arts. We advocate student exposure to these topics during secondary education because not only is this lab activity an introduction to engineering, it is insight into how many decisions are made on a daily basis across virtually all areas of life.

8. REFERENCES

[1] Richard Duda, Peter Hart, and David Stork, *Pattern Classification*, Wiley, New York, 2nd edition, 2001.

[2] Christopher M. Bishop, *Pattern recognition and machine learning*, Springer, New York, 1st edition, 2006.

[3] JJ Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of*, vol. 79, no. April, pp. 2554–2558, 1982.

[4] Patrick K. Simpson, Ed., *Neural Network Applications*, IEEE Press, Piscataway, NJ, 1st edition, 1997.

[5] P Chan and S Stolfo, "Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection," *Proc. of the Fifteenth National Conference on Artificial Intelligence*, vol. AAAI-98, pp. 164–168, 1998.

[6] Kou-yuan Huang, Senior Member, and Wen-lung Chang, "A Neural Network Method for Prediction of 2006 World Cup Football Game," *Proc. of International Joint Council on Neural Networks*, pp. 259–266, 2010.

[7] M Egmont-Petersen, D. de Ridder, and H. Handles, "Image processing with neural networks—a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279–2301, Oct. 2002.

[8] Eibe Frank, Mark Hall, Len Trigg, Geoffrey Holmes, and Ian H Witten, "Data mining in bioinformatics using Weka.," *Bioinformatics (Oxford, England)*, vol. 20, no. 15, pp. 2479–81, Oct. 2004.

[9] G. Holmes, A. Donkin, and I.H. Witten, "WEKA: a machine learning workbench," *Proceedings of ANZIIS '94 - Australian New Zealnd Intelligent Information Systems Conference*, vol. ANZIIS-94, pp. 357–361, 1994.

[10] IH Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and SJ, "Weka: Practical machine learning tools and techniques with Java implementations," *ICONIP/ANZIIS/*, vol. 1999, 1999.

[11] William M. Rand, "Objective Criteria for the Evaluation of Methods Clustering," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846– 850, 1971.

[12] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *roceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds. 1967, vol. 233, pp. 281–297, University of California Press.

[13] Steven Essinger, Ryan Coote, Pete Konstantopoulos, Jason Silverman, and Gail Rosen, "REFLECTIONS AND MEASURES OF STEM TEACHING AND LEARNING ON K-12 CREATIVE AND PERFORMING ARTS STUDENTS," *Proc. of American Society of Engineering Education*, vol. AC 2010-20, pp. 1–19, 2010.